MINUTES OF THE 41ST COMPUTER RESOURCES INTEGRATION MANAGEMENT MEETING

14 May 1991

BMDOTIC

TELEDYNE
BROWN ENGINEERING

Cummings Research Park • Huntsville, Alabama 35807

PLEASE R
SDI TECHNICAL

U2971

1/16/98 11:48:02 AM

Accession Number: 2971
Title: Minutes of the 41st Computer
Resources Integration Management
Meeting, 14 May 1991
Contract Number: DASG60-87-C-0042
Corporate Author or Publisher: Teledyne Brown Engineering,
Huntsville, AL 35807
Publication Date: May 14, 1991
Pages: 00200
Comments on Document: Contains copies of briefing
presentations. See U2909 for 39th
Meeting Minutes; U2908 for 40th.
Descriptors, Keywords: SDS Computer Resource Management
Parallel Process PPG SDI NTF NTB
NTBJPO Test NOS Network Simulation
BM C3 EV88 EVPA RTE PCS ARCSIM Model
Reengineer Argonne SEE

MINUTES OF THE 41ST COMPUTER RESOURCES INTEGRATION MANAGEMENT MEETING

14 May 1991

PREPARED FOR:

UNITED STATES ARMY STRATEGIC DEFENSE COMMAND

CONTRACT: DASG60-87C-0042

PREPARED BY:

TELEDYNE BROWN ENGINEERING
CUMMINGS RESEARCH PARK
HUNTSVILLE, ALABAMA 35807

PREPARED ON: 14 May 1991

Minutes of the 41st CRIM

1. The 41st CRIM was held 14 May 1991 at 1300 hours in Room 1C1600 at USASDC in Huntsville, Alabama. The theme for this CRIM was the second of three meetings devoted to Parallel Processing (PP). Presenters were from both the government and the private sector, and they provided topic discussions on current research and development projects using parallel processing. The meeting began with Mr. Frank Poslajko presenting the agenda. Next, Mr. Poslajko presented a slide concerning the various committee and working group meetings in the near future. During the presentation of the forthcoming meetings, SEI Contractor Software Capabilities Workshop was discussed and Dr. Ron Green commented that contractors will be required to demonstrate a Level 2 on the SEI rating scale and prepare a plan for transitioning to Level 3. In addition, a slide was presented on the different topics of parallel processing (i.e., PP computers, software and tools, compilers, etc.). This slide established a high level breakdown of the various PP subcomponents. Finally, Mr. Poslajko presented a status report on the USASDC Ada Workshop which began March 25.

2. Ms. Tina Powell of Vanguard Research presented the SDIO parallel processing activities, tools, etc. Ms. Powell began with an overview of the Parallel Programming Group (PPG). She presented the history, purpose, accomplishments, goals, and current activities of the PPG. Next, Ms. Powell discussed the May 1991 PPG meeting topics with the major theme being " The Use of Parallel Processing in SDI Simulations." Ms. Powell presented the issues from the PPG planning session. In addition, Ms. Powell discussed the threat tracking standard problem set which is planned for distribution to the PPG members in mid-July to allow common medium for comparison. Finally, Ms. Powell presented information about the next PPG meeting which will be held at Hughes Research Labs in Malibu, California, on October 29-30, 1991. The theme will be "Parallel Programming Technology Transfer within the SDIO."

3. Mr. John Hawk of the Strategic Defense Command presented for CPT Emily Andrews of the NTBJPO. He discussed the current parallel programming efforts at the National Test Facilities (NTF) which included hosting the Parallel Programming Group meeting on 30 April - 1 May. Finally, Mr. Hawk discussed the BBN TC2000 training at the NTF.

4. Mr. John Schwacke of GRC presented applications of parallel processing in the test environment system. Mr. Schwacke began with a background of the Test Environment System (TEVS). Next, he discussed how test articles are linked to the TEVS system to allow testing. Mr. Schwacke presented the different applications of parallel processing in TEVS (i.e., Course-Grain Parallelism, Fine-Grain Parallelism, and Parallel Processing Architecture). In addition, Mr. Schwacke discussed the Network Operating System (NOS) and its application and benefits as well as the advantages of vectorization in computational intensive functions. Finally, Mr. Schwacke presented NWEM distributed emulation capabilities and the composition of each node.

5. Mr. William Jarvinen of TRW presented the distributed and parallel processing in EV88/EVPA. He began with a discussion of the objectives of SDI BM/C^3 and how EV88/EVPA relates as an experimental prototype test bed for BM/C^2 processing. Next, Mr. Jarvinen discussed the Experimental Version (EV) system and how the BM/C^2 test articles are distributed on a heterogeneous network. He discussed the need for objects to follow fundamental rules during distributed operations, and the need for an appropriate tool set. In addition, Mr. Jarvinen presented an overview of the Run-Time Executive (RTE) and the interface to the process and test bed environment. Finally Mr. Jarvinen presented the Process Construction System (PCS), and he concluded with a discussion of the lessons learned in the EV project.

6. Mr. Gordon Bate of Optimization Technology Inc. presented and demonstrated the USASDC-developed ARCSIM program package. Mr. Bate began with an overview of the architecture of the Advanced Research Center (ARC). Next, he discussed the functional flow of the ARCSIM package from scenario generation and translation to data reduction and results presentation. The translator gives Network II.5 code which can be executed on a VAX, CRAY, or PC. In addition, Mr. Bate presented some of the additional functions available in ARCSIM. Finally, Mr. Bate gave a demonstration of ARCSIM showing the ease of representing a system in the package and the type of data which can be extracted once the system has been run. As of this time, new computer systems have to be modeled manually and incorporated into the library, but one of the functions for the next version is the ability of ARCSIM to gather its own data from a target computer and develop a model using this data. Dr. Davies showed interest in having models made of the computers at the Simulation Center.

7. Mr. Evan Lock of Computer Command and Control Company presented re-engineering existing software into distributed applications. Mr. Lock began by discussing the challenges of re-engineering to allow for the advantage of distributed architectures. He discussed the overall approach of re-engineering along with a summary of his company's re-engineering system. Next, Mr. Lock discussed in detail the distributed application workbench and the three main tools (i.e., Simulator, Builder, and Manager). Finally, he presented examples using the distributed application workbench and concluded with a brief summary of the advantages of this system.

8. Mr. Gregory Chisholm of Argonne National Laboratory presented SDIO0-related activities at Argonne. Mr. Chisholm began with a discussion of the Software Engineering Environment (SEE) and tools. He discussed access to parallel machines at ACRF, parallel programming classes in Ada FORTRAN & C, and the development of portable parallel programming tools. Next Mr. Chisholm discussed parallel simulation. Finally, Mr. Chisholm discussed fault-tolerant, reliable, portable computing for the SDS.

9. The meeting was adjourned by 1500. The 42nd CRIM is scheduled for 11 June 1991. This will be the third and final CRIM on parallel processing.

41st Computer Resources Integration Management Meeting
14 May 1991
List of Attendees

Name	Organization	Telephone	Fax
Dr. Davies	CSSD-TD	895-3520	
Frank Poslajko	CSSD-SP	955-1995	955-3985
Pete Cerny	CSSD-SP	955-3069	
Ted Allen	TBE	726-1285	726-1033
Robert Ellis	TBE	726-2748	726-1033
Bill Burrows	SFAE-SD-GBR-E	955-5877	955-1867
Dr. Michael Walker	SEIC/GE-HSV	883-1170 x1304	
Dr. Ron Green	SFAE-SD-GST-D	722-1844	
Steve Risner	CSSD-SA-BT	955-3848	
James Butler	ASAT	722-1078	
Les Pierre	SDIO/SDA	(703)693-1826	(703)693-1700
Mike Mitrione	DRC	(703)521-3812	(703)521-4123
Gordon Bate	OTI	721-1288	837-9682
Bettie Upshaw	CSSD-SA-BT	955-3704	
John Hawk	CSSD-NT-LO	955-3920	
John Schwacke	GRC	922-1941	
Tina Powell	Vanguard Research	(703)934-6300	(703)273-9398
Evan Lock	CCCC	(215)854-0555	(215)854-0665
Gregory Chisholm	ANL	(708)739-6235	
Jeff Craver	CSSD-SO	955-1695	
TomNuttall	CSSD-TE-P	955-3909	
Dave Gazaway	CSSD-SP	955-5209	
Nancy Byrd	CSSD-SL	955-1610	
Michael P. Gately	CSSD-SA-BE	955-4945	
Susan Roberts	PM ADCCS	895-4475	895-3178

41st Computer Resource Integration Management (CRIM) Meeting
Open Action Items

1.	Provide a status update on the software organization and development at NTF.	John Hawk - 955-3920
2.	Schedule status briefing on SDS committees to include purpose, accomplishments, plans, and schedules.	Frank Poslajko - 955-3920
3.	Establish a data reduction planning committee.	Barbara Rogers 722-1518
4.	DISC 4 to coordinate with Ada 9X project office on Ada language deficiencies.	Bob Johnson AV227-0259
5	ADCCS project office to report on the number of Ada waiver requests submitted to DISC 4.	Denise Jones 895-3397
6.	Broaden distribution of SESE Specification Document to CASE developers.	Frank Poslajko - 955-3920
7.	Schedule a Rational & ISI tools demonstration.	Frank Poslajko - 955-3920
8.	Ensure the SDI TIC is placed on the USASDC documentation distribution list.	Frank Poslajko - 955-3920
9.	Determine requirements/procedures necessary to incorporate models for the Simulation Center's computers in ARCSIMs model library.	Frank Poslajko - 955-3920
10.	Develop process for technical data transfer of SDC developed products to the SDS TIC.	Frank Poslajko - 955-3920 Mike Mitione (703)521-3812
11.	What is the status of the TC2000 Ada compiler at the NTB.	John Hawk - 955-3920

PRESENTER: FRANK POSLAJKO

1) Agenda
2) Action Items
3) Computer Resource Meetings

41st CRIM Meeting

14 May 91

COMPUTER

RESOURCES

INTEGRATION

MANAGEMENT

MEETING

THEME: PARALLEL PROCESSING

Agenda
41st Computer Resources Integration Management Meeting
14 May 1991
Conference Room 1C1600, 0800-1500 Hours

0800-0810	Introduction	Frank Poslajko 955-1995
0810-0825	SDIO Parallel Processing Activities, Tools, etc.	Tina Powell (703) 934-6300
0825-0845	National Test Bed Software Development Parallel Processing Functions & Issues	CPT Andrew (719) 380-3265 John Hawk 955-3920
~~0845-0905~~	~~Advanced Research Center Architecture Overview~~	~~Bob Cooley 955-4360~~
0905-0935	Network Operating System, Applications & Test Environment	John Schwacke - GRC 922-1941
0935-1000	Real-time BM/C3 Applications of Parallel Processing	Bill Jarvinen - TRW Michael Gately 955-4945
1000-1010	Break	
1010-1035	Computer Resource Simulation Tool for Distributed Systems (ARCSIM)	Gordon Bate - OTI 721-1288
1035-1110	Re-engineering Existing Applications in Distributed Systems	Evan Lock (215) 854-0665 Computer Command & Control
1110-1140	Parallel Processing Facilities, Research and Services at Argonne National Lab	Greg Chisholm (708) 972-6815
1140-1300	Lunch	

(Continued)

Agenda
41st Computer Resources Integration Management Meeting
14 May 1991
Conference Room 1C1600, 0800-1500 Hours

1300-1305	Action Items	Frank Poslajko 955-1995
1305-1320	SDIO Parallel Processing Activities, Tools, etc.	Dr. Leslie Pierre (702) 693-1826
1320-1335	National Test Bed Software Development Parallel Processing Functions & Issues	CPT Andrew (719) 380-3265 John Hawk 955-3920
~~1335-1350~~	~~Advanced Research Center Architecture Overview~~	~~Bob Cooley 955-4360~~
1350-1405	Network Operating System, Applications & Test Environment	John Schwacke - GRC 922-1941
1405-1420	Real-time BM/C3 Applications of Parallel Processing	Bill Jarvinen - TRW Michael Gately 955-4945
1420-1435	Computer Resource Simulation Tool for Distributed Systems (ARCSIM)	Gordon Bate - OTI 721-1288
1435-1450	Re-engineering Existing Applications in Distributed Systems	Evan Lock (215) 854-0665 Computer Command & Control
1450-1500	Parallel Processing Facilities, Research and Services at Argonne National Lab	Greg Chisholm (708) 972-6815

41st Computer Resources Integration Management Meeting
Action Items

1.	Provide a status update on the software organization and development at NTF.	John Hawk 955-3920
2.	Schedule status briefings on SDS committees to include purpose, accomplishments, plans, and schedules.	Frank Poslajko 955-1995
3.	Establish a data reduction planning committee.	Barbara Rogers 722-1518
4.	DISC 4 to coordinate with Ada 9X project office on Ada language deficiencies.	Bob Johnson AV227-0259
5	ADCCS project office to report on the number of Ada waiver requests submitted to DISC 4.	Denise Jones 895-3397
6.	Broaden distribution of SESE Specification Document to CASE developers.	Frank Poslajko 955-1995
7.	Schedule a Rational & ISI tools demonstration.	Frank Poslajko 955-1995
8.	Establish process for transfering technical data on projects briefed to SDS TIC.	Poslajko / Mitrione 955-1995 / (703) 521-3812

Computer Resources Meetings
(After 9 Apr 91 CRIM)

DATE	MEETING
• 25 Mar - 3 Jun 91	Ada Workshop (80 Hours) @ USASDC, Huntsville, AL
• 30 Apr - 1 May	SDIO Parallel Programming Group (PPG) Meeting @ NTB, Colorado Springs, CO
• 30 Apr - 2 May	SEI Contractor Software Capability Evaluation Workshop @ SEI, Pittsburg, PA
• 9 May 91	Ada Executive Overview (1 Hour) @ USASDC, Huntsville, AL
• 13-15 May 91	Software Quality Evaluation Special Committee Meeting (Rewrite of SDS Software Standards, Practices, and Conventions Document) @ ARINC Research, Colorado Springs, CO
• 15-16 May 91	Algorithmic Architecture Program (A2P) Review @ TRW, Huntsville, AL
• 30-31 May 91	SEE Software Engineeging Environment Committee Meeting @ DRC, Crystal City, VA
• 22-24 July 91	SDS CRWG Meeting @ TBE, Huntsville, AL

Parallel Processing Topics Matrix

PP Computers GT PFP (KDEC) FTP (Draper Labs) TC 2000 HYPERCUBE (INTEC iPSC/8600) ALLIANT APTEC	**PP Software** 6-DOF DPSIM	**Organization** SDIO SDC A/F Navy National Labs Universities	**Applications** OBDP BM/C3
Classifications SISD - Solbourne-5/501 SIMD - DAP 510 MISD MIMD iPSC/860 INTEL Alliant/FX8	**SW Tools**	**Projects/Programs** A2P DINC	**Networks** HPAN HYPERCHANNEL (50 Mbps) HYPERBUS (10 Mbps) LAN WAN
Architecture RISC CISC Shared/Distributed Memory	**Languages** Ada FORTRAN C LINDA Crystal	**PP Requirements** • Memory/Throughput • Applications	**Connectivity** RS-232 ETHERNET T1 (1.54 Mbps) T5 Satellite
Operating Systems NOS POSIX UNIX	**Compilers** GT Ada Compiler	**PP Issues** • PP Tools Lacking • Ada Compiler Not Mature • Little Ada Code Available	**Protocols** GOSEP TCP/IP
Utilization		**Training** ARGONNE NAT LAB	**Signal Processor** Acousto-Optic Processing

2nd Ada Fundamentals Workshop Update

- Started 25 Mar 91 (Currently in the seventh of ten weeks)
- 8 hours/week
    - Mondays, 4 hours in the classroom
    - Fridays, 4 hours on VAX terminals solving problems
- <u>18 of the original 22 participants are actively attending the workshop.</u>

PRESENTER: Tina Powell

SDIO Parallel Processing Activities, Tools, etc.

STRATEGIC DEFENSE SYSTEM

Command & Control Element (C2E)

Parallel Programming Briefing for the Computer Resource Integration Meeting

14 May 1991

Tina Powell
Vanguard Research, Inc.

AGENDA

COMMAND
& CONTROL
ELEMENT

- PPG Overview
- FY 91 PPG Goals/Activities
- May 1991 PPG Meeting Topics
- May 1991 PPG Meeting Planning Session
- Summary

PPG OVERVIEW

PPG HISTORY

COMMAND
& CONTROL
ELEMENT

- Began as an Encore User's Group after the SDIO Purchased Seven Encores and Distributed Them Throughout the SDI Research Community
  - Promote Interaction in the SDI Research Community
  - Initial Meeting had Approximately 10 Attendees

- 4th Meeting held in April 1989
  - "SDI Parallel Computing Group Meeting" Hosted at Argonne National Laboratory in Argonne, Illinois (ANL) Under the Auspices of the SDIO Phase I Directorate
  - Sessions were held on Models for Parallel Processing, Assignment and Tracking Algorithms, Systems, and Ada and Tools for Parallel Computing

- 5th Meeting held in October 1989
  - Spiraled into the Parallel Programming Group (PPG) under the Leadership of CPT Steve Johnson (SDIO/ENA)
  - Hosted at USAF SSD in Los Angeles, California
  - First Day Focused on the System Element Programs and Their System Operation and Integration Functions (SOIF) Parallel Processing Requirements
  - Second Day Focused on Parallel Programming Research and Results

PPG HISTORY (CONTINUED)

COMMAND
& CONTROL
ELEMENT

- 6th Meeting Held in April 1990
  - Hosted by USASDC in Huntsville, Alabama
  - Focused on Parallel Processing Resources/Tools and Implementations

- 7th Meeting Held in October 1990
  - Hosted by Los Alamos National Laboratory in Los Alamos, New Mexico
  - Focused on the Performance Achievement of Parallel Processing Implementations of SDS System Functions
  - Held a Planning Session to Openly Discuss How the Group can have a more Beneficial Impact on SDI

PPG OVERVIEW

COMMAND
& CONTROL
ELEMENT

SPONSOR: SDIO/SDA (Dr. Pierre)

PURPOSE: The Purpose of These Meetings is to Explore the Extent to Which Parallel Architecture Computers Could be of Use for the Rapid Execution of SDS System Integration Functions

- Transition Technology into the Operational System
- Identify How the State of the Art can Benefit the C2E Program
- Focus on the System Integration Function Algorithms
- Forum for Sharing Common Research and Results
  - Techniques in Parallel Programming
  - Parallel Design Concepts
  - Speed-Up Results
- Advertise Available Resources and Tools for the SDI Community

FY 90 ACCOMPLISHMENTS

COMMAND
& CONTROL
ELEMENT

- Showed that a Significant Increase in Throughput is Attainable Using Parallel Programming
- Showed that Parallel Processing Provides More Compute Power for the Money
- Increased the Knowledge Level in the Community About Parallel Processing
- Demonstrated that Near Real-time Processing is Possible for a Subset of the Phase I Threat
- Demonstrated that Real-time Execution of the System Integration Functions is Within Reach

PPG FY91 GOALS

COMMAND
& CONTROL
ELEMENT

- Transition Technology From Research Programs to Element Programs
- Develop a Standard Problem Set
- Assess How Parallel Programming Can Aid in Simulation Development
- Assess the Current State of the Practice in Implementing the System Integration Functions
- Start to Acquire a Data Base for Building Confidence in Achieving the System Integration Function Requirements

CURRENT PPG ACTIVITIES

COMMAND
& CONTROL
ELEMENT

- Baseline Problem Set Definition Underway
- Executive Committee For Dissemination of Results
- Newsletter to Facilitate Communications
- Focus on GPALS Architecture
- Becoming More Product Oriented

MAY 1991 PARALLEL PROGRAMMING GROUP (PPG) MEETING TOPICS

MAY 1991 PPG MEETING

COMMAND
& CONTROL
ELEMENT

- Major Theme: "The Use of Parallel Processing in SDI Simulations"
- Held at the National Test Facility to Assist in Technology Transfer
- Held a Planning Session to Discuss Development of a Baseline Problem Set and MOPs/MOEs for the PPG to Use in Their Research Efforts

MAY 1991 PPG MEETING TOPICS

COMMAND
& CONTROL
ELEMENT

Simulation Topics

- Distributed Simulation of a Satellite Attack/Defense Engagement Model (SADEM)
- SDI Simulation with a Heterogeneous Computer Architecture
- Simulations on Massively Parallel MIMD Computers
- Parallel and Distributed Simulation
- Communication Simulation
- EV88/EVPA

Other Topics

- Time Warp
- Parallel Programming Activities at Georgia Tech
- Multi-processor for Data Fusion
- Parallel Assignment Algorithm
- Radar Tracking on the MasPar MP-1

MAY 1991 PARALLEL PROGRAMMING GROUP (PPG) PLANNING SESSION

PPG PLANNING SESSION ISSUES

COMMAND & CONTROL ELEMENT

- Integrate PPG with Other SDIO Efforts
  - Tracking Panels
  - IS&T Efforts
  - Program Elements

- Testing
  - Common Testing Required to Compare "Apples with Apples"
  - Standard Problem Set Needed for use in Benchmarking

- MOEs/MOPs
  - Strawman to be Sent out for Review

- Dissemination of Information
  - Committee Created to Disseminate Information
  - Parallel Computing Newsletter
  - Parallel Computing Resources/Areas of Expertise Document

STANDARD PROBLEM SET

COMMAND
& CONTROL
ELEMENT

- Proposed Threat
  - Coordinate with Tracking Panels
  - Will be sent to all PPG Members by mid-July for Evaluation
  - NTF is Going to Produce an Unclassified Threat Tape
- Standard Threat Subsets
- By end of Summer, Standard Threat Should be Readied for Validation Tests
- Standard SDS Architecture Definition
- Complete Description of the Algorithms Used
- Definition of Constants
- Standard MOEs/MOPs
- Strawman of Battle Management Functions

NEXT MEETING

COMMAND
& CONTROL
ELEMENT

Theme: "Parallel Programming Technology Transfer within the SDIO"

- October 29-30, 1991
- Hughes Research Labs in Malibu, CA
- Topics
  - Where are we Today in Parallel Programming Technology?
  - What can we Achieve in the Near Future?
  - How do we Transition Technology into the Operational BM System?
- POC:

Tina Powell
Vanguard Research, Inc.
703-934-6300; 703-273-9398 (F)
e_mail: powell@jedi.sdio.mil

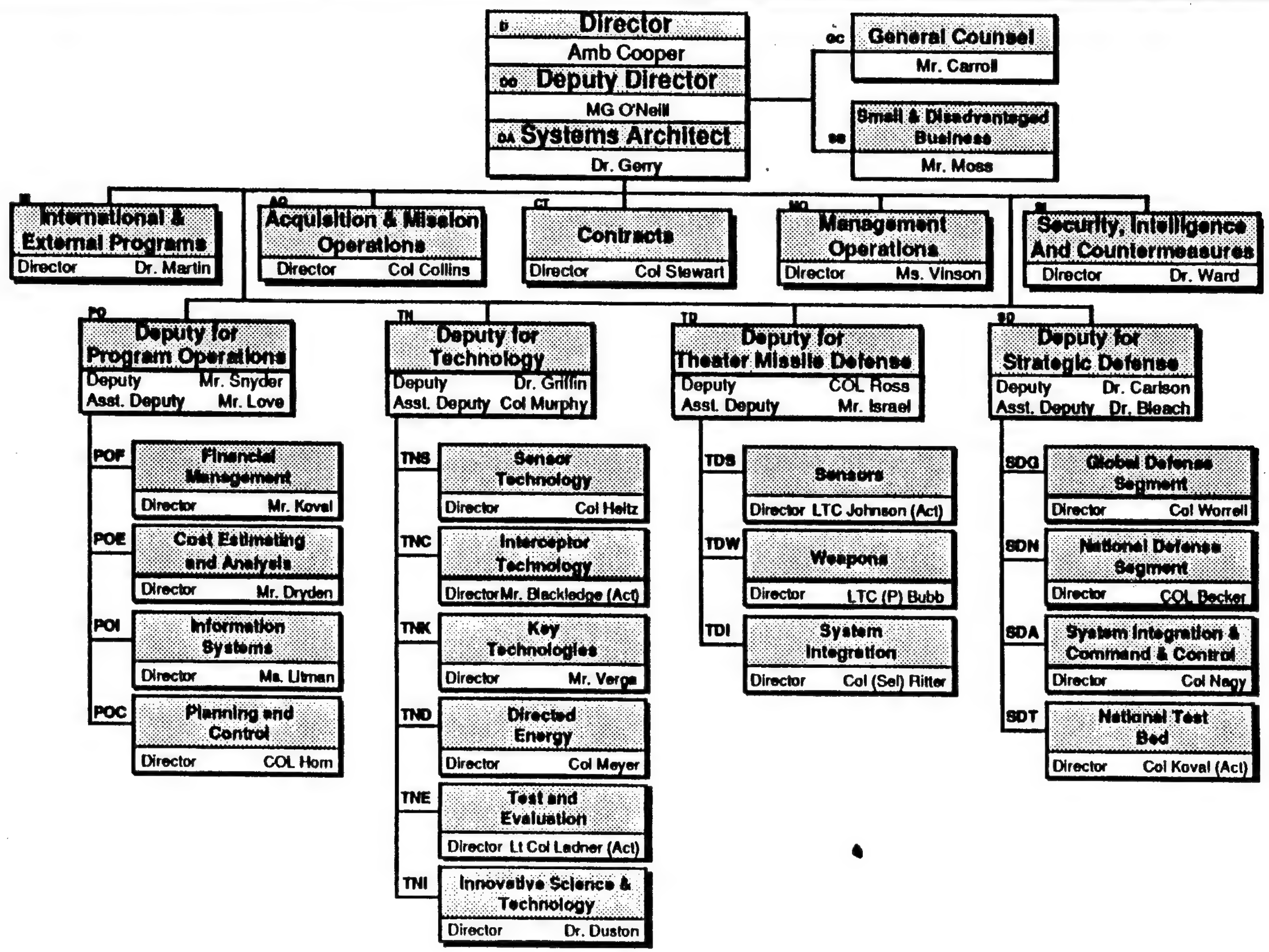
SUMMARY

PPG SUMMARY

COMMAND
& CONTROL
ELEMENT

- Provides Forum for Technology Transfer
  - IS&T
  - Program Elements
- Integrates Parallel Programming into the C2E (i.e., via the A2 Program)
- SDIO Body of Knowledge in Parallel Programming
  - Provide Support to Element Programs
  - Provide Resources and Expertise to the Community
  - Assembling Data for Building Confidence in Meeting the Processing Requirements for an SDS

STRATEGIC DEFENSE INITIATIVE ORGANIZATION



SDIO.dch 4/19/91 - 2

PRESENTER: John Hawk for CPT Emily Andrew

National Test Bed Software Development Parallel Processing Functions & Issues

John Hawk

NTF Parallel Programming Efforts

Current Parallel Programming Efforts

- Hosted SDIO Parallel Programming Group conference at NTF 30 April - 1 May
  - Two NTF presentations
    - -- "Integrated System Test Capability (ISTC) Status," Chet Murphy, MITRE Corp.
    - -- "Simulation of Very Large Problems," Jim Hardy, Geodynamics
- BBN TC2000 Training at the NTF
  - Week of 13 May, TC2000 Overview and "C" and FORTRAN courses
  - Week of 3 June, Real-Time Operating System Training

NATIONAL TEST BED

PRESENTER: John Schwacke

Application of Parallel Processing in the Test Environment System

GRC

Application of Parallel Processing in the Test Environment System

Presented To:

41st Computer Resource Integration Meeting
14 May 1991

J. Schwacke

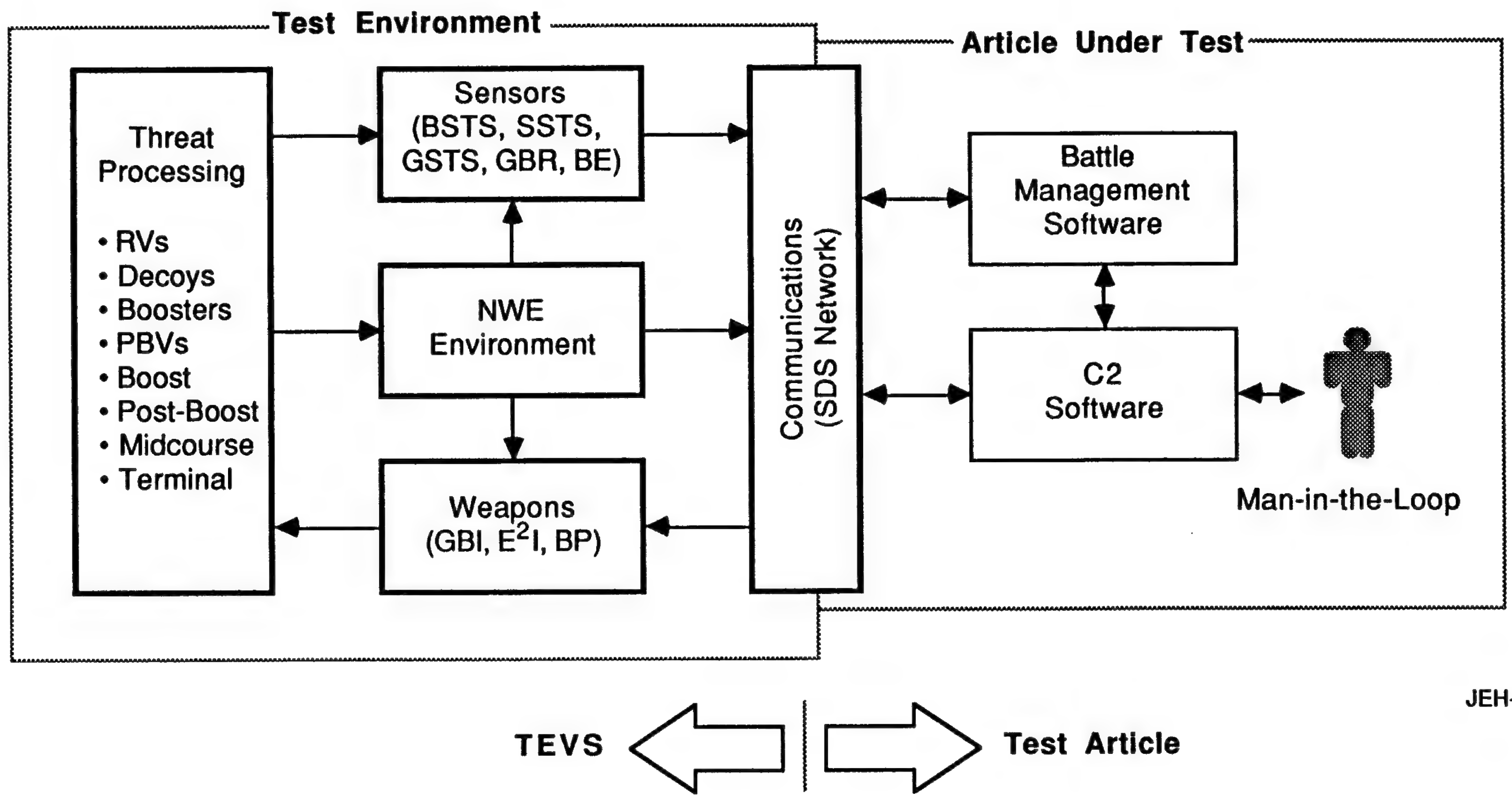
Background

- TEVS (Test Environment System) is a distributed, real-time simulation of the SDS system, designed to exercise prototype BM and C2 software
- TESSE (Test Environment Support System Enhancements) is an on-going program to improve the TEVS capability
- The TEVS system has been used in a number of experiments under the USASDC EV program
- The TEVS design makes use of both course-grain and fine-grain parallelism on Alliant computers
- TEVS software which supports distributed simulation has utility in other programs (currently being used to implement a surrogate framework for L2SS model development)

GRC

Experimental Systems Testing

Provide the input data streams for BM/C2 and respond to BM/C2 directives



JEH-17-42

GRC

Application of Parallel Processing in TEVS

Course-Grain Parallelism

Components of the simulation execute in parallel as separate tasks on multiple computers and multiple processes within a computer

Separate tasks for each sensor, threat, communications, nuclear effects, groups of weapons and farms

TEVS Capability

Network Operating System, Simulation Executive

Fine-Grain Parallelism

Each task makes use of special capabilities of the architecture

Vectorization for basic vector operations and special vectorization of heavy computations (Threat Object Propagation)

TEVS Capability

Vectorized Functions

Parallel Processing Architecture

A distributed computing system developed to support the evaluation of communications protocols

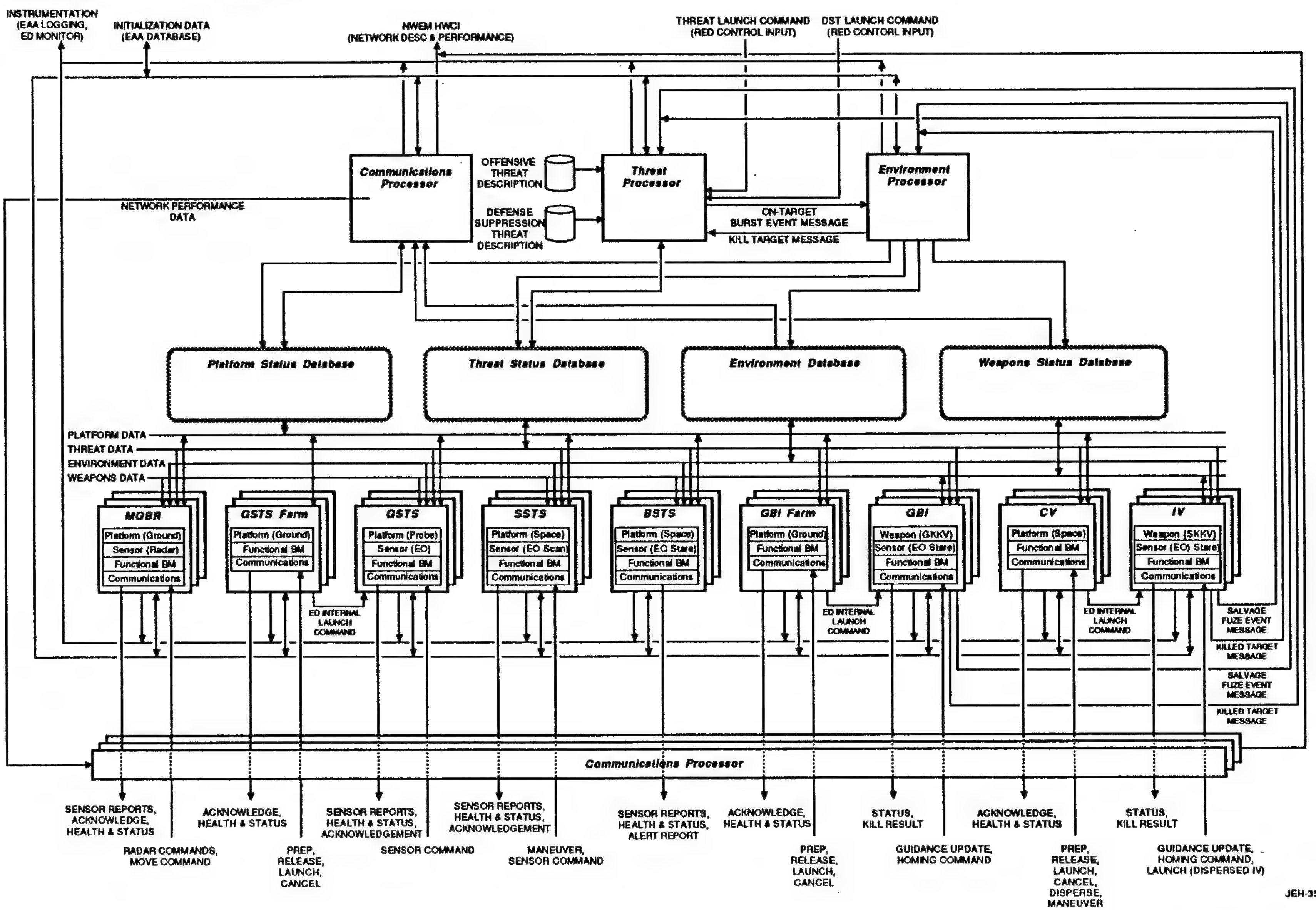
12 node system, each with 4 processors, communications via APTEC IOC-200 and Ethernet. Can be used as a real-time, distributed processing system.

TEVS Capability

Real-time distributed processing system and programming environment

GRC

Experiment Driver Configuration



JEH-35-33

Software Layers

Application Code

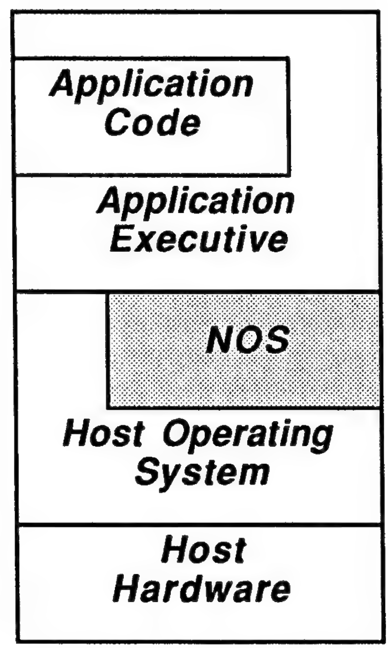
- BM/SOIF algorithms and functions
- ED SDS element models
- Interfaces only to the Application Executive
- Execution sequence is controlled by the executive

Host Operating System

- Provided by the testbed
- Supports file services
- Performs task-level scheduling
- Allocates HW resources
- Provides communications protocol between testbed computers

Host Hardware

- Provided by the testbed
- CPU
- Memory
- Disk
- Communications



Application Executive

- Real-Time Executive (RTE)
- ED Model Executive
- Controls the execution of the application code within a task
- Hides interfaces to the host OS and NOS
- Maintains event queues and performs scheduling of functions

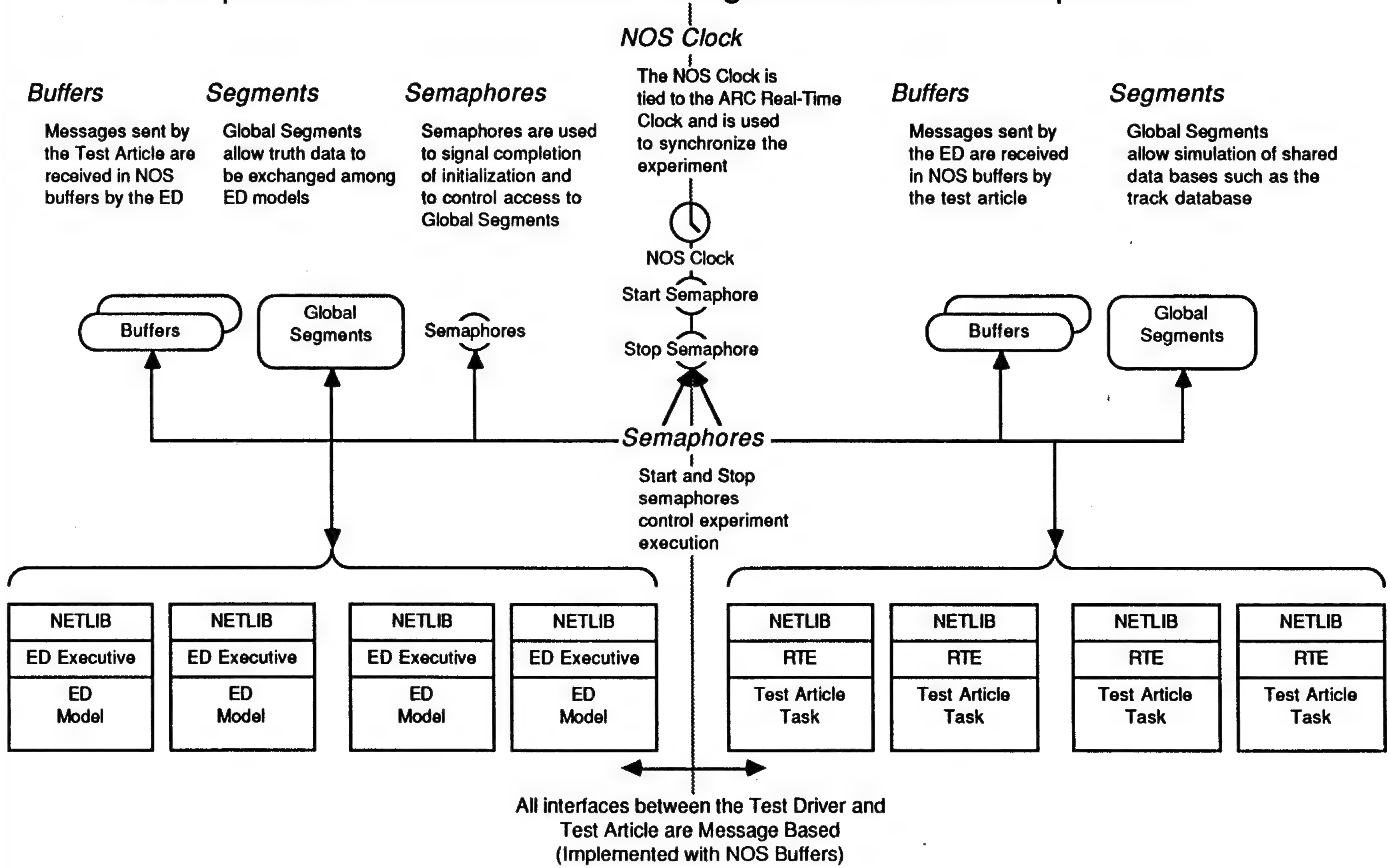
NOS

- Provides network-level abstraction for communicating processes
- Provides communications protocol between tasks
- Hides physical location of processes
- Interfaces to host communications, clock and shared memory services
- Provides services to the Application Executive
- NOS is not a stand-alone OS
- NOS does not control OS resources
- NOS does not control the application

GRC

NOS Use in EV Experiments

NOS provides the services which integrate a distributed experiment



NOS Components

NETLIB

A library of NOS functions, linked with the user's application. NETLIB provides NOS services to the user's application.

NRM

The Node Resource Manager (NRM) process controls the creation and destruction of NOS objects within each node's NOS shared section.

Mordred

The Mordred process provides clean-up services for tasks that terminate abnormally and would otherwise leave NOS in an inconsistent state.

TCP Agent

The TCP Agent process transmits and receives messages via TCP as required by NOS

NCP

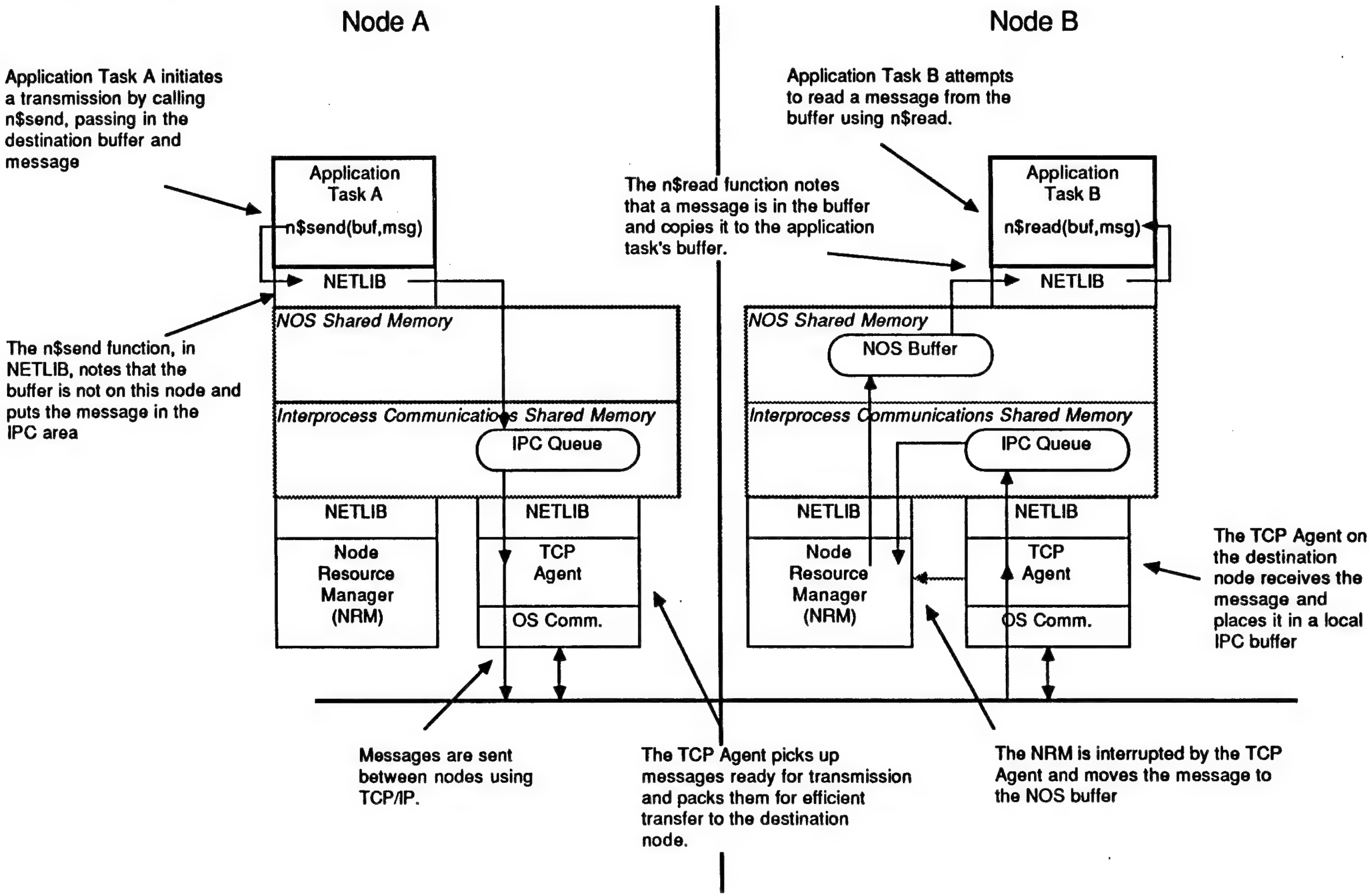
The Network Control Program (NCP) is a transient process started by the user to create, delete, modify and observe NOS buffers, semaphores, etc.

Scheduler

The Scheduler is an optional NOS process which provides task-level start, stop and resume control over NOS application tasks

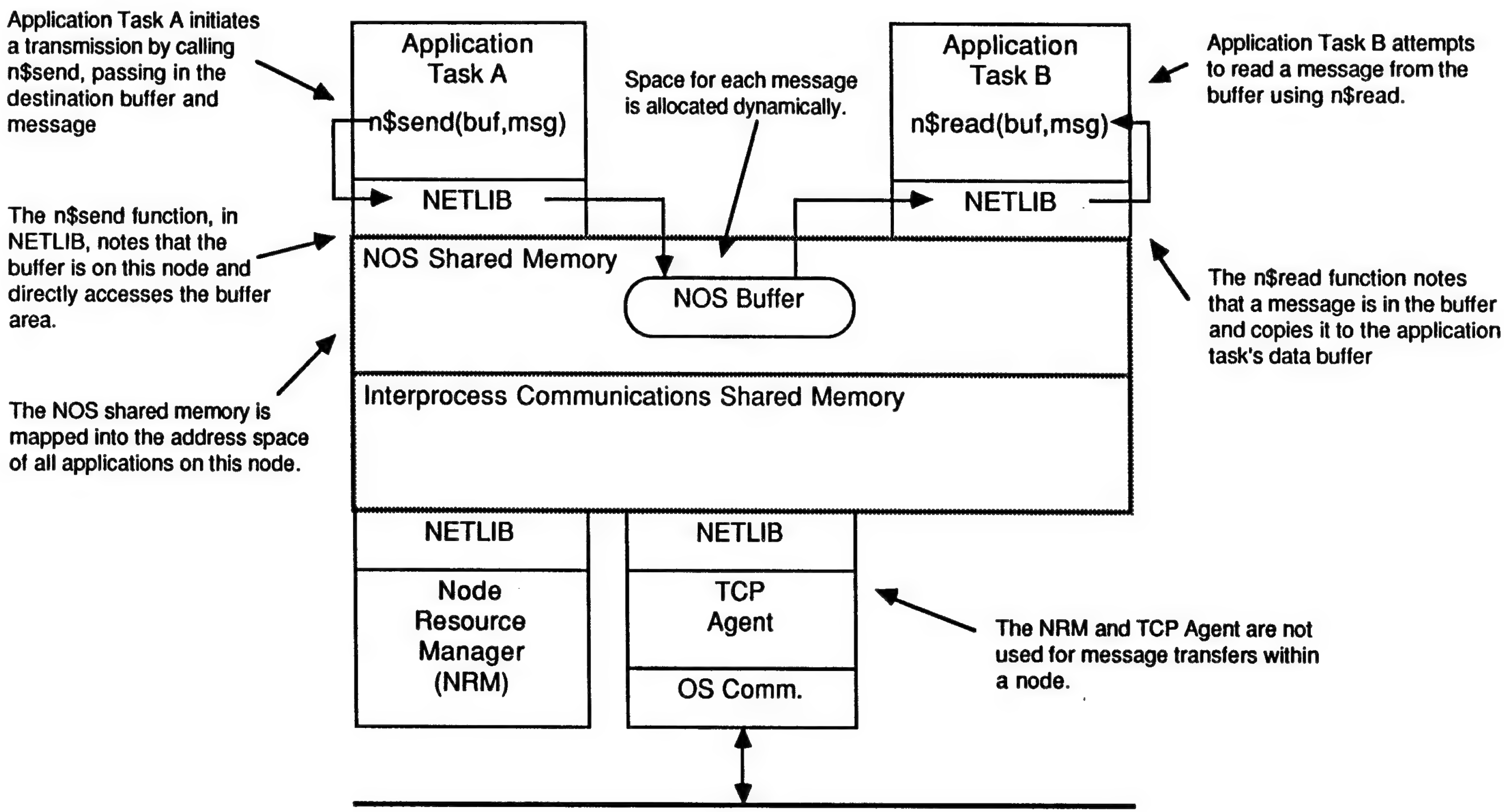
GRC

Message Flow Across Nodes



GRC

Message Flow Within a Node



GRC

Model Execution Environment

Model Environment

NOS

- Global Clock Access
- Intertask Communications
- Global Data Access
- Global Semaphores

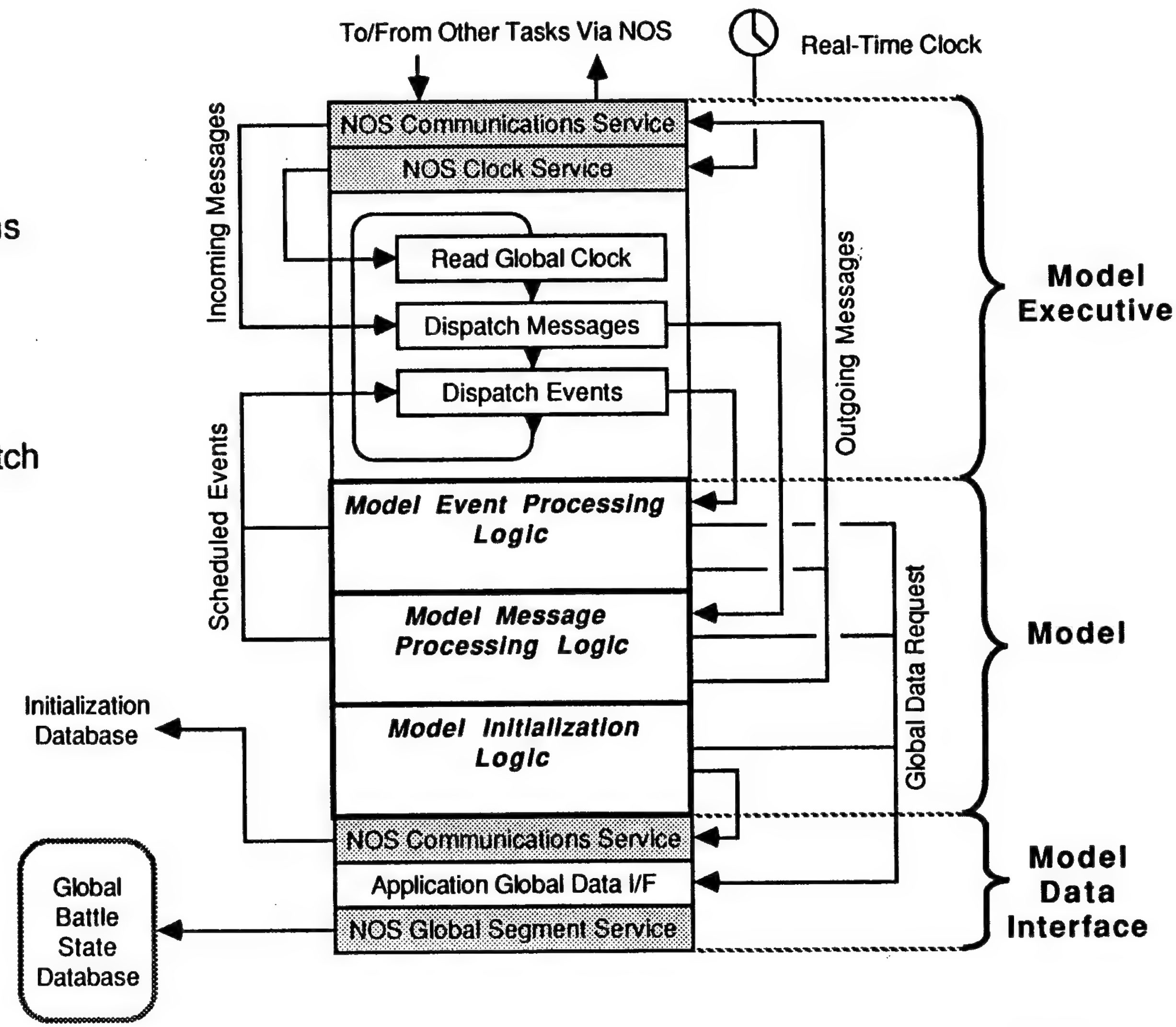
Model Executive

- Scheduling, Event Dispatch
- Message Dispatch

Model Data Interface

- Application Global Data
- Initialization Data

"The Model" is a discrete event simulation under the control of the model environment.



JEH-21-39

GRC

Vectorization of Compute-Intensive Functions

For large threats and small step sizes, the processing associated with threat object propagation becomes significant

Techniques for restructuring the trajectory integration process (RK4) to take advantage of vectorization

Method

No Vectorization

```
For each object
   k1 = hf(xn,yn)
   k2 = hf(xn+h/2,yn+k1/2)
   k3 = hf(xn+h/2,yn+k2/2)
   k4 = hf(xn+h,yn+k3)
   yn1 = yn + (k1+2k2+2k3+k4)/6
```

Vectorization

```
For each object
   k1 = hf(xn,yn)
For each object
   k2 = hf(xn+h/2,yn+k1/2)
For each object
   k3 = hf(xn+h/2,yn+k2/2)
For each object
   k4 = hf(xn+h,yn+k3)
For each object
   yn1 = yn + (k1+2k2+2k3+k4)/6
```

Propagation code written in FORTRAN, compiled with the vectorizing compiler. An Ada shell was written to interface the FORTRAN-generated object to the Ada-based threat process

Performance of Vectorized Trajectory Function

1,000 objects flown from time 800 to time 1200 in 2 second steps

Measured minimum, maximum, and average time per step

Results

	Minimum	Average	Maximum
Pure FORTRAN	0.223 s	0.226 s	0.232 s
Ada with FORTRAN Trajectory Integration	0.440 s	0.445 s	0.470 s
Pure Ada	1.330 s	1.366 s	1.425 s

10,000 objects flown from time 800 to time 1200 in 2 second steps

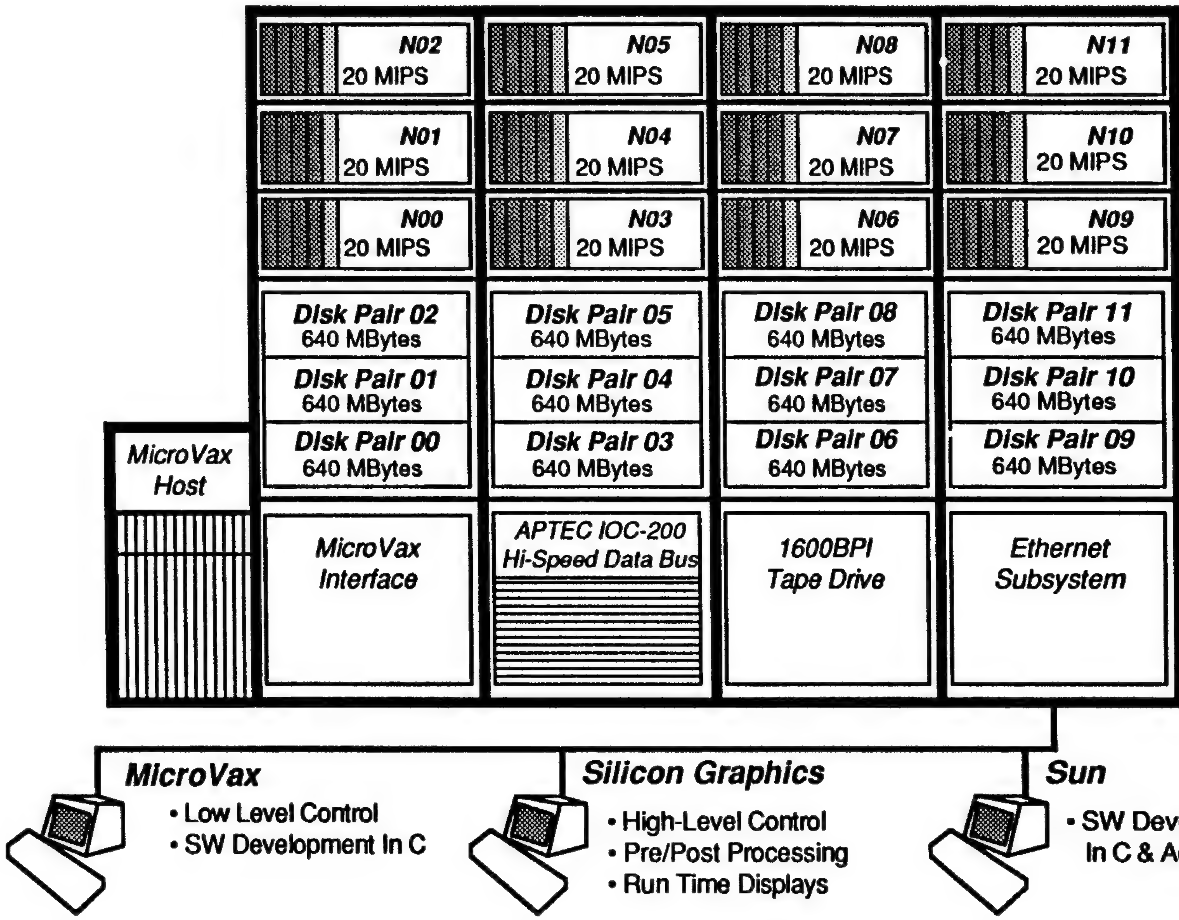
Measured minimum, maximum, and average time per step

Results

	Minimum	Average	Maximum
Pure FORTRAN	2.431 s	2.432 s	2.464 s
Ada with FORTRAN Trajectory Integration	4.710 s	4.814 s	6.530 s
Pure Ada	13.580 s	13.891 s	14.435 s

GRC

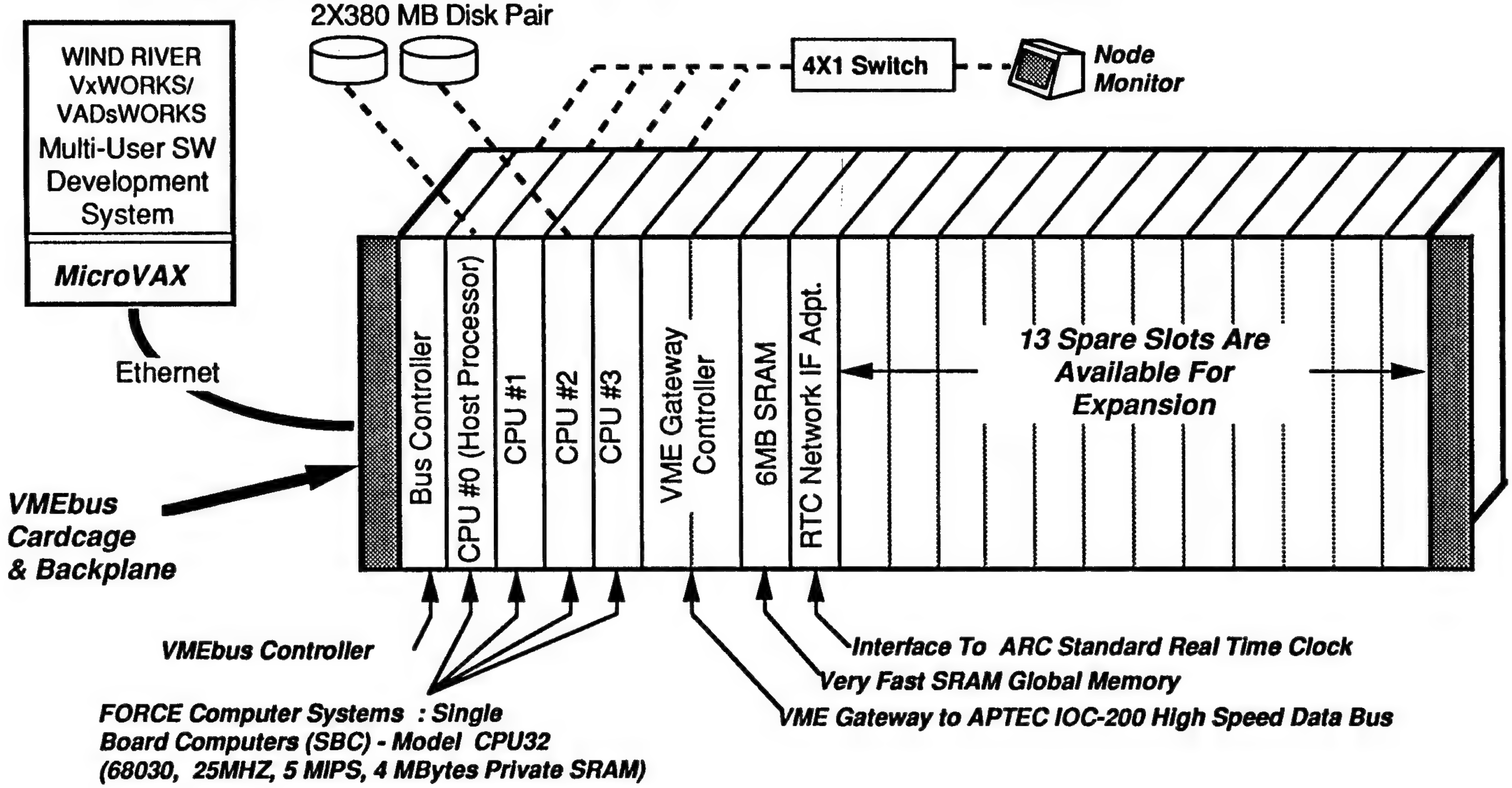
NWEM Distributed Emulation Capabilities



- 12 - 20 MIP Test Nodes (Currently)
- 4 Separate 5 MIP FORCE 68030 Processors Per Node (w/4 MBytes Static RAM)
- Real Time Clock Allows All CPUs on All Nodes To Use Common Time (4usec 'tick', Pause, Reset, etc.)
- Industry Standard High-Speed VMEbus Backplane
- VAX-based Software Cross-Development Systems Supports C, C++, and Ada Based Implementations from VAX and Sun
- Real-Time Multiprocessor OS
- 12 Disk Pairs Allow On-Node Data Recording with No Bus Loading (2-320MB, 16 MS)

GRC

Node Composition



- Separate 68030-Based Single Board Microprocessors
- Host Microprocessor for Executive, I/O, Local Physical Layer
- Industry-Standard High-Speed VMEbus Backplane (Expandable, Upgradable)
- VAX-Based Software Cross-Development System Supports C, C++, & Ada-Based Implementations
- VXWorks Real-Time Multiprocessor (UNIX Clone) Operating System

CWC-910204Gen-3b

Capabilities and Plans

Simulation Executive

- Currently using the distributed model executive in real-time runs
- Implemented and testing two distributed discrete event simulation controllers (Modified Chandy Misra null message and Misra marker message schemes)

NOS

- NOS Version 8.14 currently operational at ARC
- Supports Alliant FX-8 (Unix) and VAX 8800 (VMS) systems and currently uses TCP/IP for processor-to-processor communications
- Version 8.20 to be released for use 15 June
- New version supports Alliant FX-8 (Unix), VAX 8800 (VMS), and Silicon Graphics (Unix) systems includes support for HPAN (High Performance ARC Network) in addition to TCP/IP

Distributed Processing System (NWEM)

- 12 node system complete and operational in GRC's facility including Ada and C development environment and real-time operating system

PRESENTER: William A. Jarvinen

Distributed and Parallel Processing in EV88/EVPA: An Experimental Prototype Testbed for BM/C2 Processing

TRW EVPA
Harris
Alpha Tech
MindGate
SPARTA

Distributed and Parallel Processing in EV88/EVPA: An Experimental Prototype Testbed For BM/C2 Processing

W. A. Jarvinen

14 May 1991

Distributed and Parallel Processing In EV88/EVPA

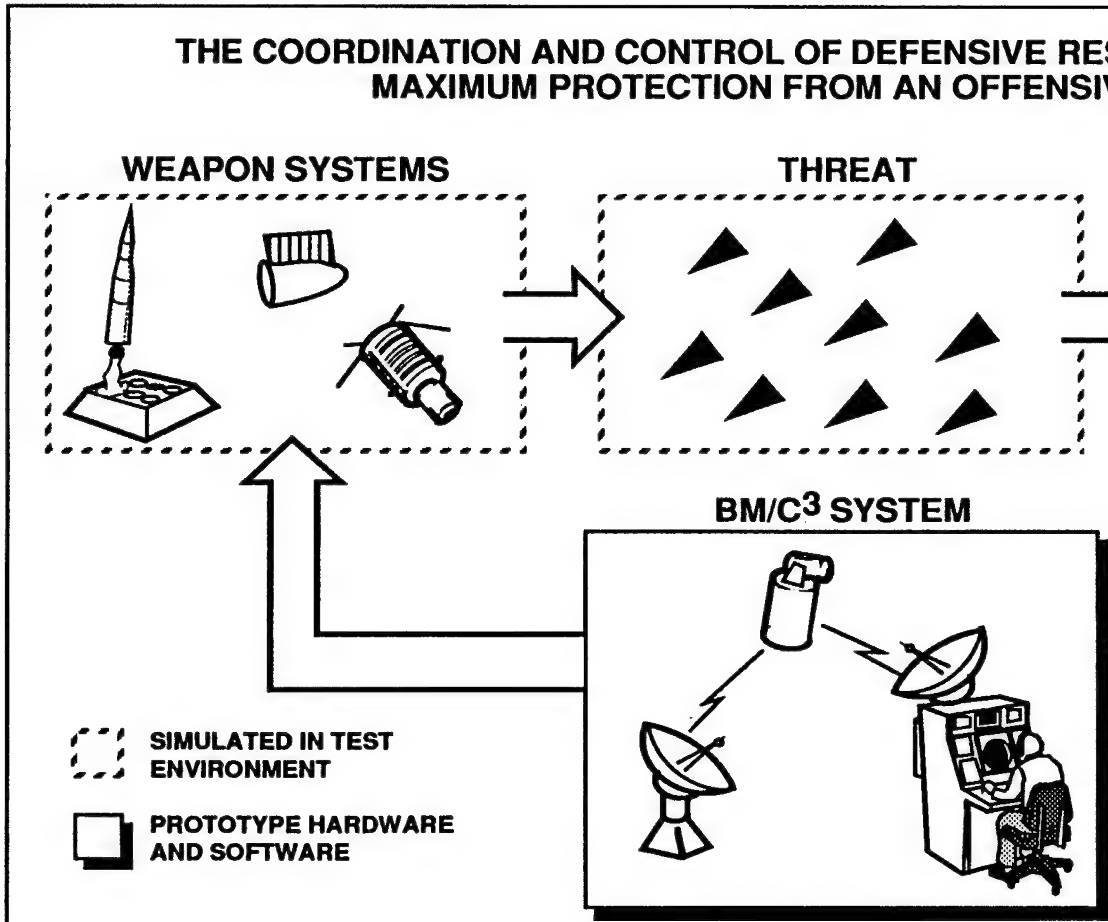
TRW

Harris
Alpha Tech
MindGate
SPARTA

- What Is EV88/EVPA
- Functions Represented
- Distribution Of Functions To Hardware
- Distributed And Parallel Operations Of The Test Article
- Implementation Of Distributed And Parallel System
- Lessons Learned
- EV88/EVPA Accomplishments

SDI BM/C^3 OBJECTIVE

TRW
Harris
Alpha Tech
MindGate
SPARTA



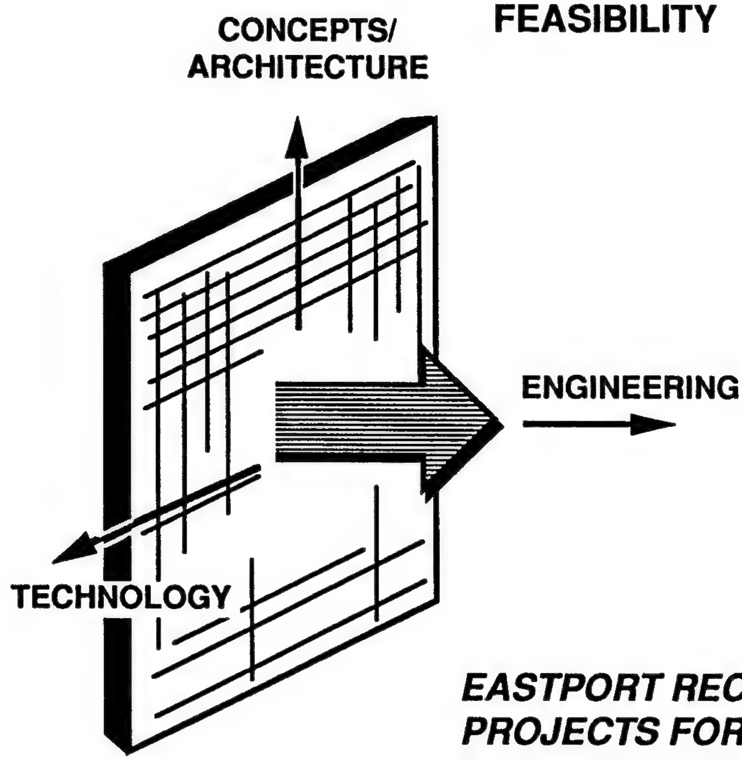
MAJOR QUESTIONS

- CAN THE SOFTWARE BE BUILT?
- CAN THE SYSTEM BE TESTED?
- WHAT IS THE SYSTEM PERFORMANCE ?

0491-040/01

EV88/EVPA: AN EXPERIMENTAL PROTOTYPE TESTBED FOR BM/C2 PROCESSING

BATTLE MANAGEMENT PROTOTYPE TESTING PUSHES BM CONCEPTS AND TECHNOLOGY TOWARD ENGINEERING FEASIBILITY

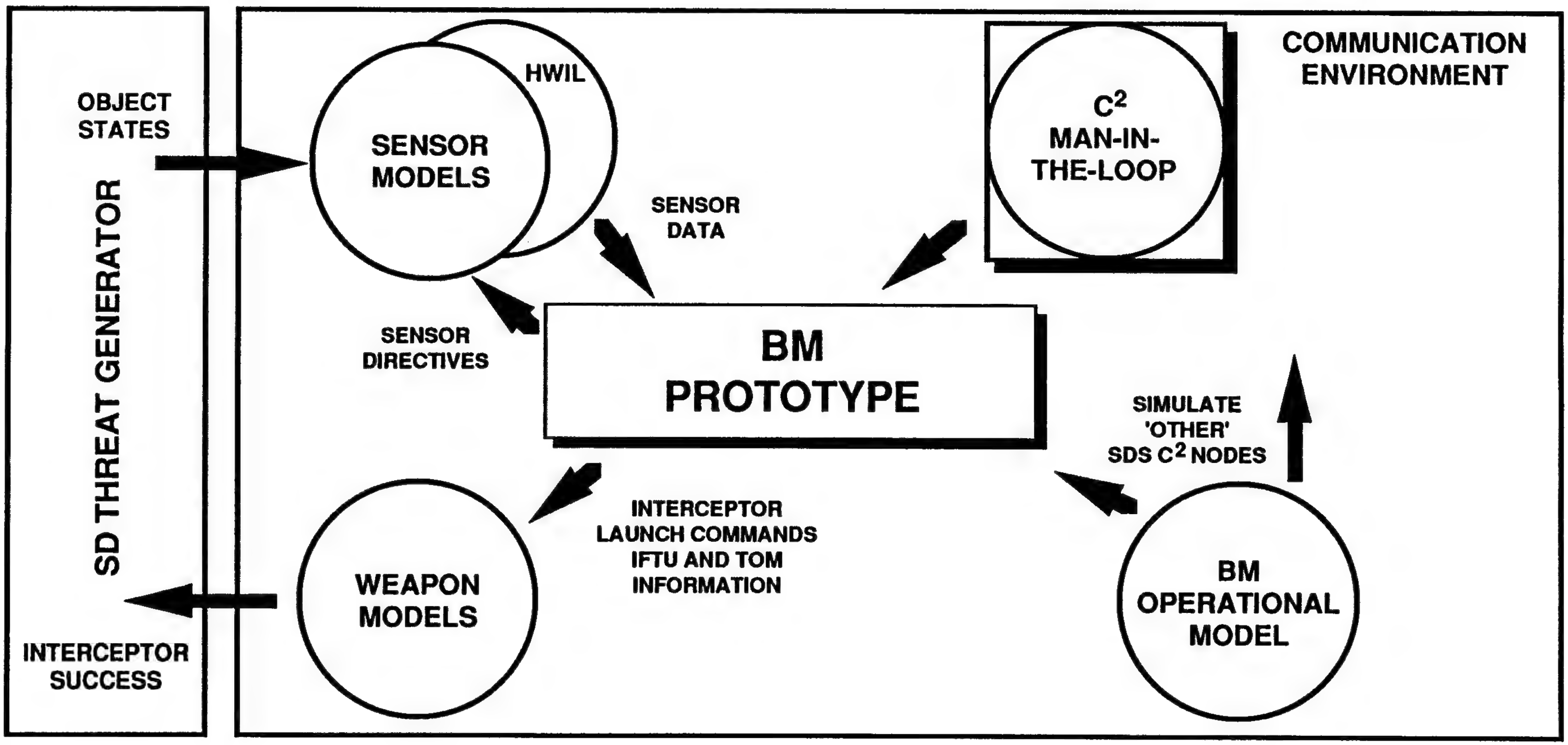


- EARLY PROTOTYPE BM SOFTWARE - NOT A SIMULATION
- DYNAMIC BATTLE PLAN EXECUTION AND DISTRIBUTED SYSTEM MANAGEMENT
- REAL TIME MIL, HWIL CAPABILITY
- MIL STD SOFTWARE ENGINEERING AND DEVELOPMENT

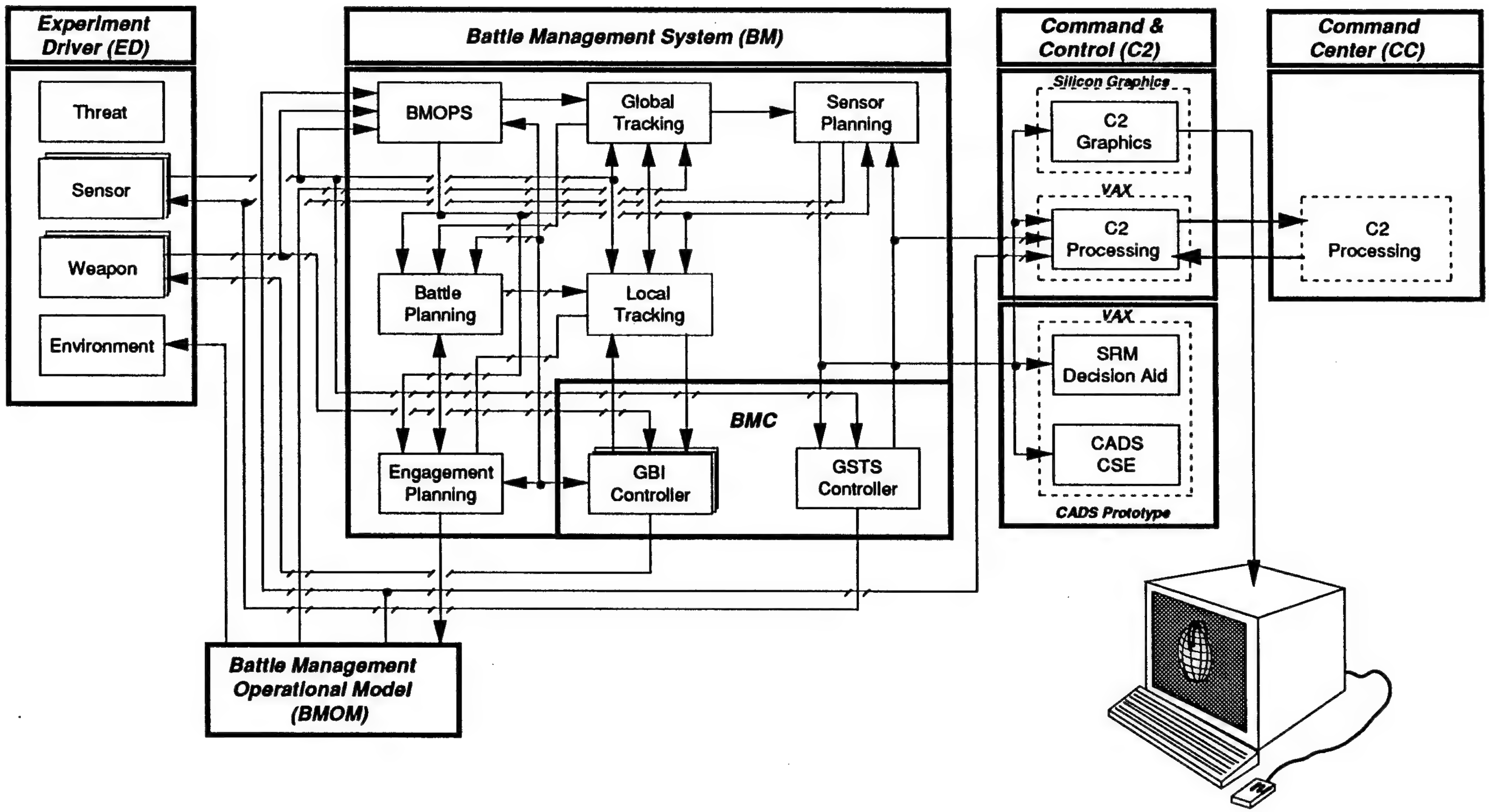
EASTPORT RECOMMENDATION: "...SUPPORT PROTOTYPING PROJECTS FOR THE DEVELOPMENT OF BM SYSTEMS... MUST EXHIBIT PROPERTIES IMPORTANT TO SDS, SUCH AS RESPONSIVENESS, ROBUSTNESS, NETWORK BASED, TESTABILITY..."

THE BM PROCESSING TESTBED APPROACH

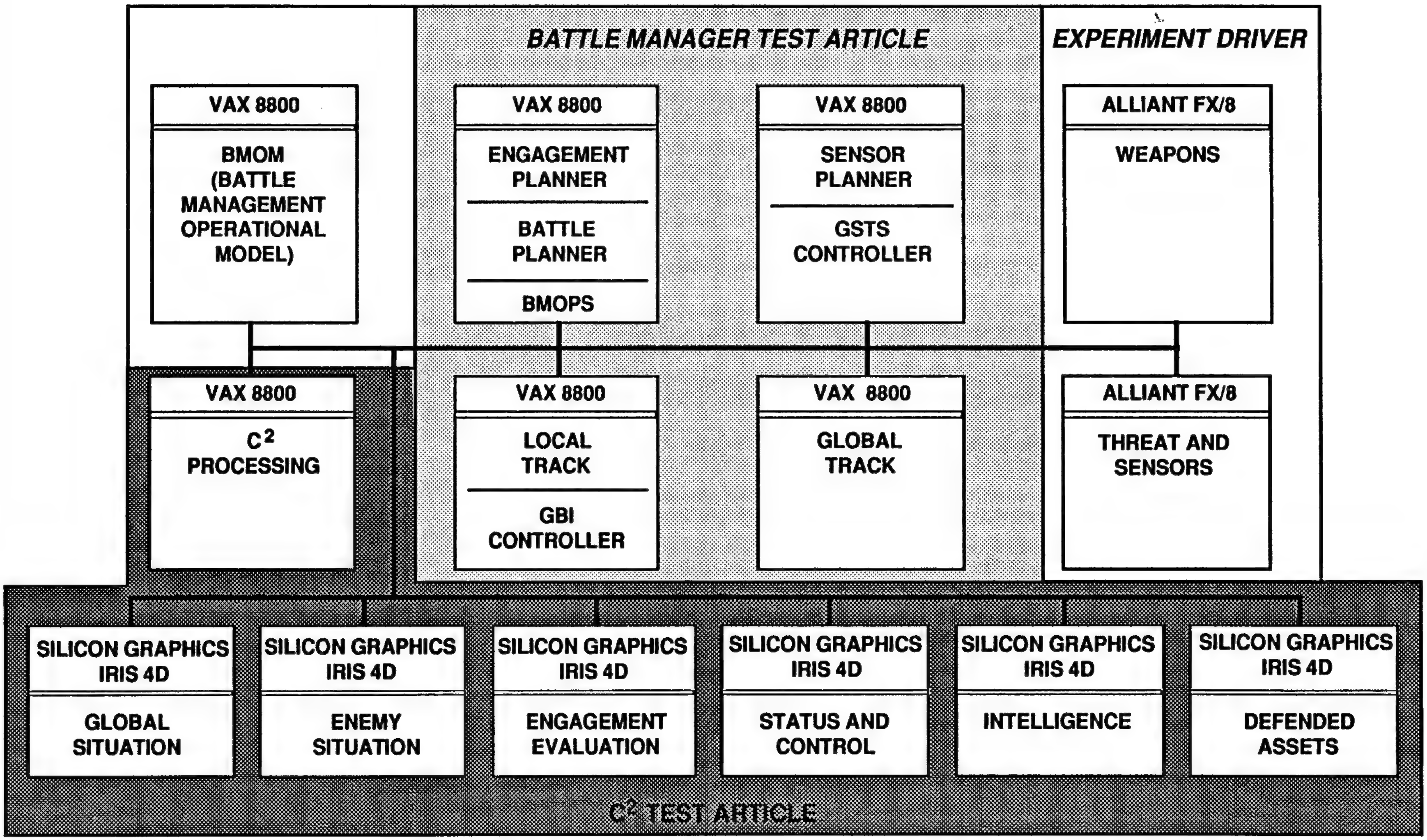
BUILD A "BM PROTOTYPE" AND PLACE IT ON AN SDS NODE
EMBED THE NODE IN THE SYSTEM DATA STREAM



EV BM/C3 Experimental System



BM/C2 TEST ARTICLES DISTRIBUTED ON HETEROGENEOUS NETWORK



EVPA BATTLE MANAGER IS A S/W PRE-PROTOTYPE

TRW
Harris
Alpha Tech
MindGate
SPARTA

GLOBAL TRACK (GT)

- Reassess Cluster Threat
- Delete Tracks
- Process Midcourse Data
- Process Thrusted Data
- Assess Cluster Threat
- Update CADS Tracks
- Track Thrusted Objects*
- Track Clusters*
- Update Cluster Tracks
- Process Lethal Assignments
- Process Modeled Data*
- Process CADS Data

BATTLE MANAGEMENT OPERATIONS SUPPORT (OP)

- Update Health and Status
- Process Nuclear Events
- Process SDS Data
- Process CADS Data

LOCAL TRACK (LT)

- Delete Tracks
- Discriminate Objects *
- Assess Object Threat
- Reassess Object Threat
- Assess Object Kill
- Process MGBR States
- Track Objects *
- Process Object States
- Process CADS Data
- Update Object Tracks
- Update Lethal Tracks

SENSOR PLANNER (SP)

- Process Thrusted Track Data
- Aggregate Clusters
- Allocate Sensors
- Generate Pointing Parameters
- Process Cluster Tracks
- Process Defense Assets
- Process GSTS Status
- Process CADS Data

MIDCOURSE ENGAGEMENT PLANNER (ME)

- Generate Engagement Plans
- Setup Engagement Planning
- Process status report
- Process CADS Data

MIDCOURSE BATTLE PLANNER (MB)

- Setup Battle Planning
- Allocate Weapons
- Process Status Report
- Process Engagement Plans
- Process CADS Data

GBI CONTROLLER (GB)

- Process Engagement Plan
- Process In-Flight Status
- Process Launch Response
- Update Health and Status
- Compute IFTU
- Monitor Health and Status
- Load Object Tracks
- Launch Interceptor
- Compute TOM
- Process CADS Data

GSTS CONTROLLER (GS)

- Update Health and Status
- Monitor Health and Status
- Process Launch Resonse
- Generate Pointing Commands
- Generate Launch Commands
- Process CADS Data

** Functional Model*

Distributed Operations Require that an Object Follow Fundamental Rules

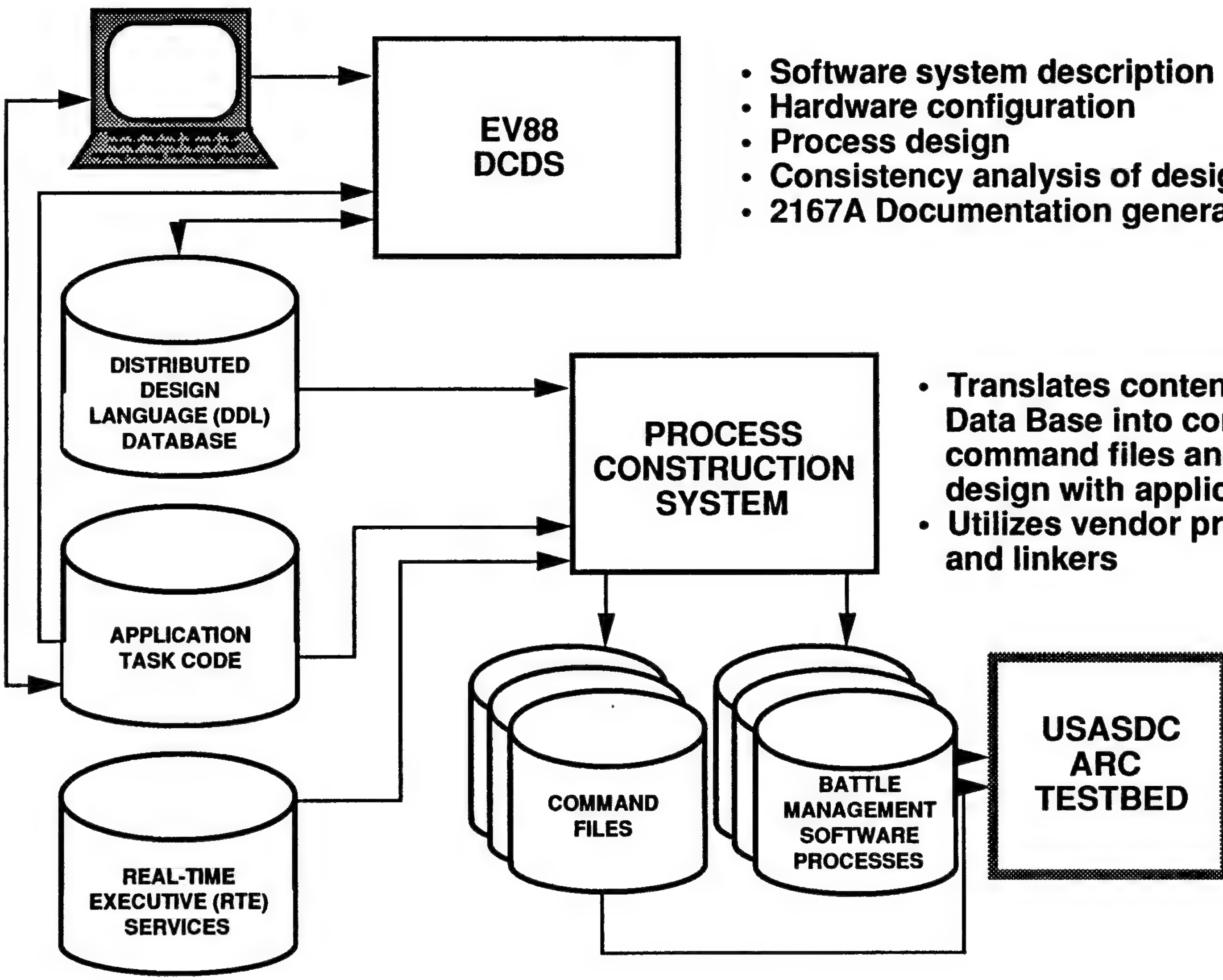
TRW
Harris
Alpha Tech
MindGate
SPARTA

- It Must Communicate via Messages with Other Processes through Well-Defined Interfaces and Communications Protocols.
- It Must have its Own 'Thread of Control' which Models an Independent Activity Capable of Stand-Alone Execution if Provided with the Correct Input Data
- It Should be Possible to Construct a Library of Distributable Components that Does Not Rely on Specific Network Details– This Guarantees the Ability to Distribute the Processes at Run-Time
- Support the Definition of Processes to Enable Dynamic Creation of Instances of that Process via Data-Driven Methodology

Distributed Operations Require An Appropriate Tool Set

- DCDS/DDL
- Run-Time Executive
- Process Construction
- Distributed Ada Environment Tools

DCDS Support For Distributed Battle Management Design



- Software system description
- Hardware configuration
- Process design
- Consistency analysis of design
- 2167A Documentation generation

- Translates contents of Design Data Base into control tables, command files and binds design with application code
- Utilizes vendor provided compilers and linkers

- Real-time distributed environment
- BM/C 3 experimentation

91-04-00500

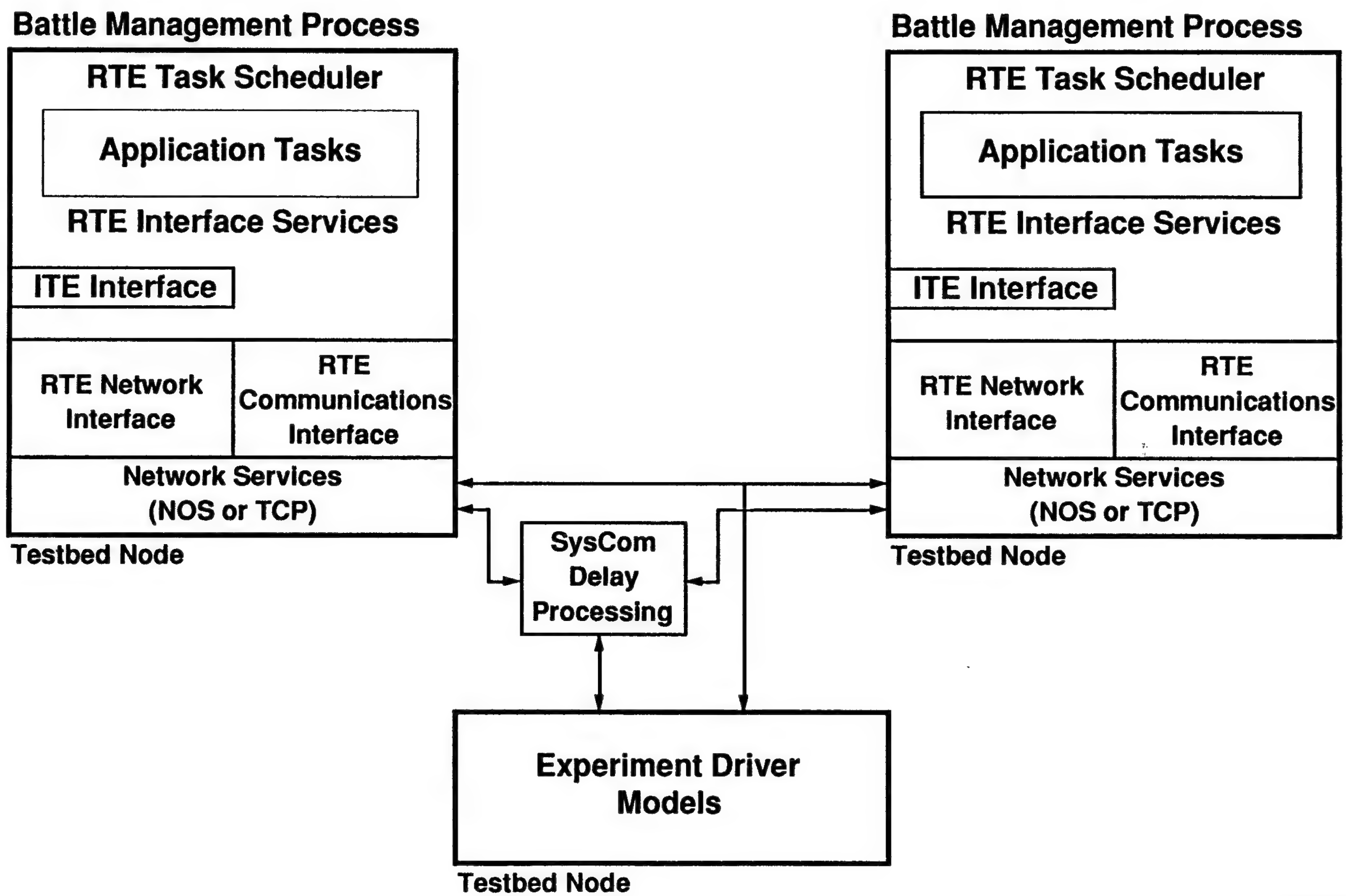
Run-Time Executive (RTE)

Provides Control and Services for Distributed Ada Application Software

- Process Initialization
- Network Connectivity
- Task Scheduling
- Message and Buffer Services
- Event and State Services
- Clock Event Services
- File (Local and Global) Services

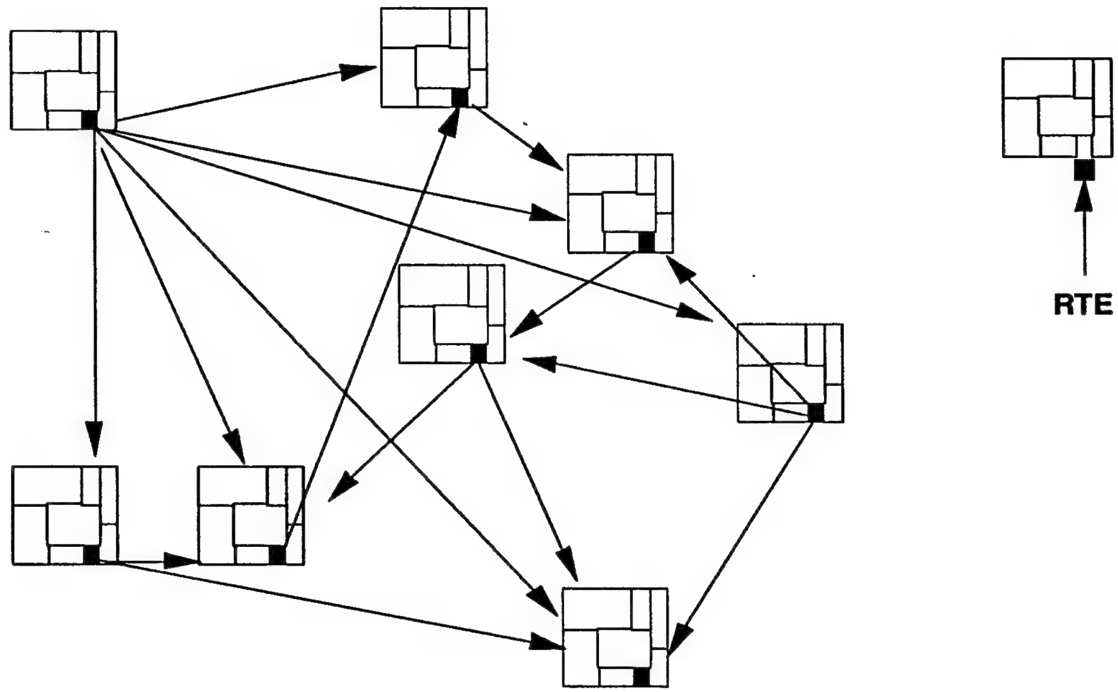
Run-Time Executive's Interface to the Process and Testbed Environment

Harris
Alpha Tech
MindGate
TRW
SPARTA



EV91-04-0505

Typical Process Connectivity



Run time connectivity provides a generic capability to build the interfaces to the number of TLCSC processes within a BM experiment configuration.

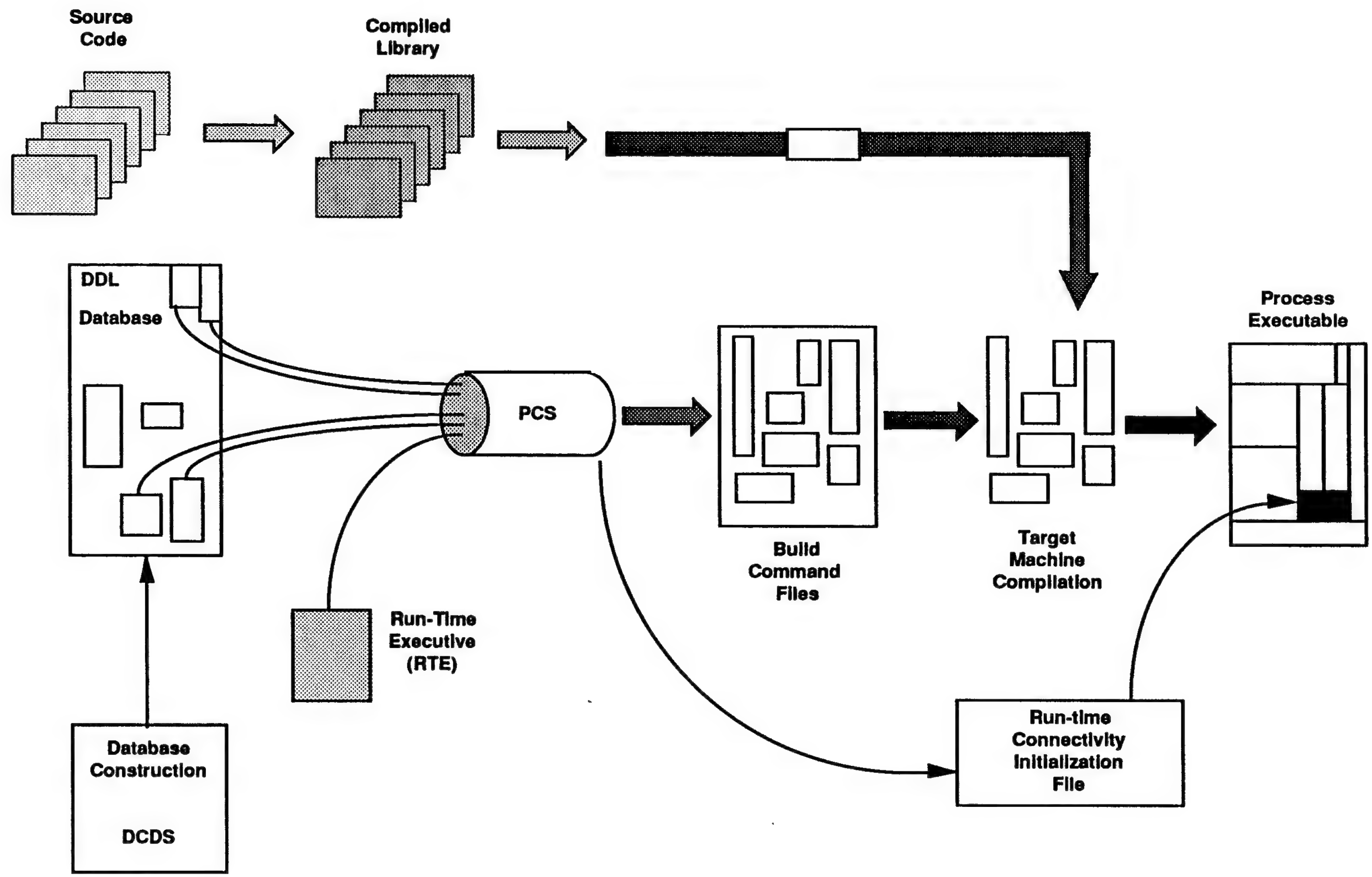
- The DDL and associated code is integrated and tested once.
- Allows for relationships of the test article to be dynamic at run time (multiple midcourse engagement planners each assigned to a class of controllers).

The Process Construction System (PCS)

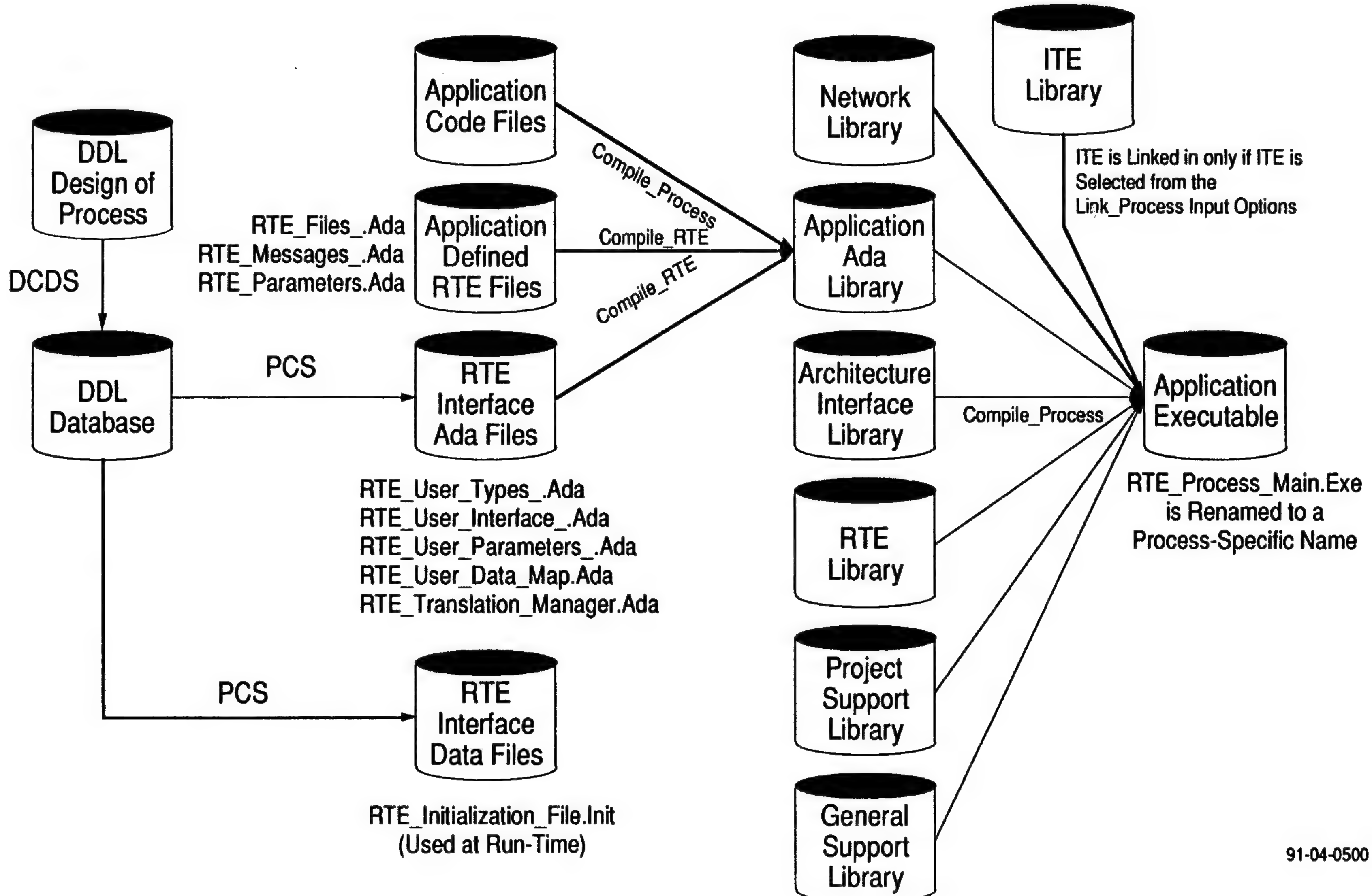
Harris
Alpha Tech
MindGate
TRW
SPARTA

- PCS Builds a Distributed Interface to the Process-Specific Files, Messages, Buffers, etc. Through the DCDS Database
  - Creates Ada Process-Specific Code
  - Creates Process-Specific Initialization Files
- User Provides Application Tasks and Other Source Code Libraries and Packages
- PCM Ensures the Consistency of the Constructed Source Code and Initialization Files to Produce a Distributable Environment

Process Construction



Building A Run-Time Environment



91-04-0500

Distributed Ada Environment Tools

- Distributed Design Language (Ada Compatible)
- Validation of Architecture Interface and Consistency
- Construction of a Distributed System
- Execution and Experiment Generation of a Distributed System
- Automated Logging via the Integration Test Environment (ITE) and the Run-Time Executive (RTE)
- Analysis Tools for Inspection of Run-Specific Data
- Capability to Playback or Regenerate Subsystem and Standalone Environments for Problem Isolation

LESSONS LEARNED ON EV

TRW
Harris
Alpha Tech
MindGate
SPARTA

- Real-Time Processing
  - End-To-End Algorithm implementation in Real-Time Frequenly Results in "Answers" Very Different Than Indicated by Standard Event-Based Simulations
  - Real-Time Insertion of Weapon/Sensor Releases,Strategies,Tactics By User/Operator Personnel Will Resolve HIC Issues and Defining New Requirements Which Cannot be Anticipated By a Simulation Environment
- Distributed Systems
  - Distributed Works - for Real-Time,For Ada, For Large-Scale Developments
  - Requires The Development Of An Appropriate Tool Set
  - Have To Make The Investment In Tools

LESSONS LEARNED (Continued)

TRW

Harris
Alpha Tech
MindGate
SPARTA

- Ada Code Development
  - Ada Works Well For Distributed And Real-Time
  - Large Scale Integration Of Multiple CSCIs Developed by Multiple Contractors Worked
  - Ada Metrics
- Defining A Stable Set Of Interface Definitions And Message Definitions For All Elements - Neither Simple Nor Straight-Forward For Complex Dstributed Real-Time Systems

> - The real benefits in prototyping are in getting there - not just the end product.
> - Prototyping reduces risk prior to FSD by making your mistakes early in the product development cycle.

EV88/EVPA ACCOMPLISHMENTS

TRW
Harris
Alpha Tech
MindGate
SPARTA

<u>WE HAVE BUILT A COMPLETE BM PROTOTYPE</u>

- A BM NODE EMULATION WITH "REAL WORLD" DATA DRIVEN INTERFACES TO A PERCEIVED ENVIRONMENT
  - <u>NOT</u> An End-To-End System Simulation
- PERFORMS A COMPLETE SET OF BM FUNCTIONS
  - Track File Data Management
  - Threat Assessment
  - Weapon Target Assignment
  - Generate Engagement Solution
  - Provides Interceptor Post-Commit Support
  - Kill Assessment
- OPERATIONAL I/F DEFINITIONS WITH WEAPONS AND SENSORS (BP,BE,GSTS,GBR,GBI,HEDI,E2I)
- REAL-TIME INTERACTIVE DRIVEN BY SENSOR AND HUMAN-IN-CONTROL INPUTS
- MANY MATURE ALGORITHMS - MODULAR DESIGN (Plug-In/Plug-Out)
- HAD TO BE DONE DISTRIBUTED AND PARALLEL: NO OPTIONS AVAILABLE
- LESSONS LEARNED ON DISTRIBUTED AND PARALLEL, TOOLS DEVELOPED AND PROBLEMS SOLVED SHOULD TRANSFER AS "GRAIN SIZE" DECREASES

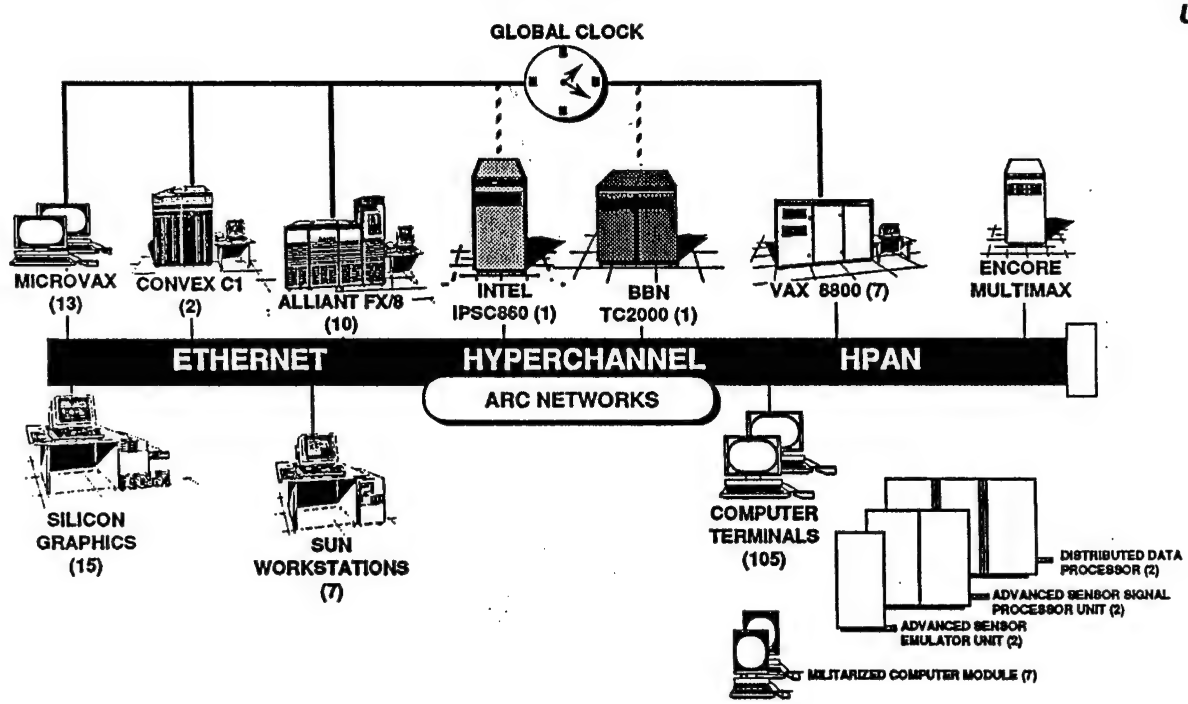
PRESENTER: Gordon Bate

Computer Resource Simulation Tool for Distributed Systems (ARCSIM)

GORDON BATES, OTI

ADVANCED RESEARCH CENTER

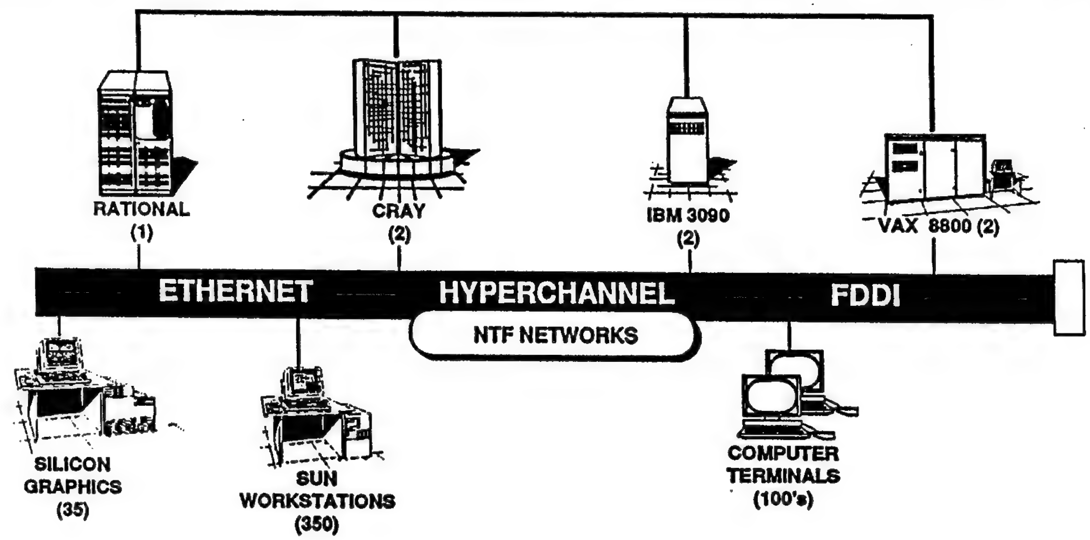
UNCLASSIFIED



SUPPORT COMPUTER PERFORMANCE PREDICTION FOR DISTRIBUTED EXPERIMENTS USING MULTIPLE NETWORKS, PROTOCOLS, COMPUTERS, AND APPLICATIONS

NATIONAL TEST BED

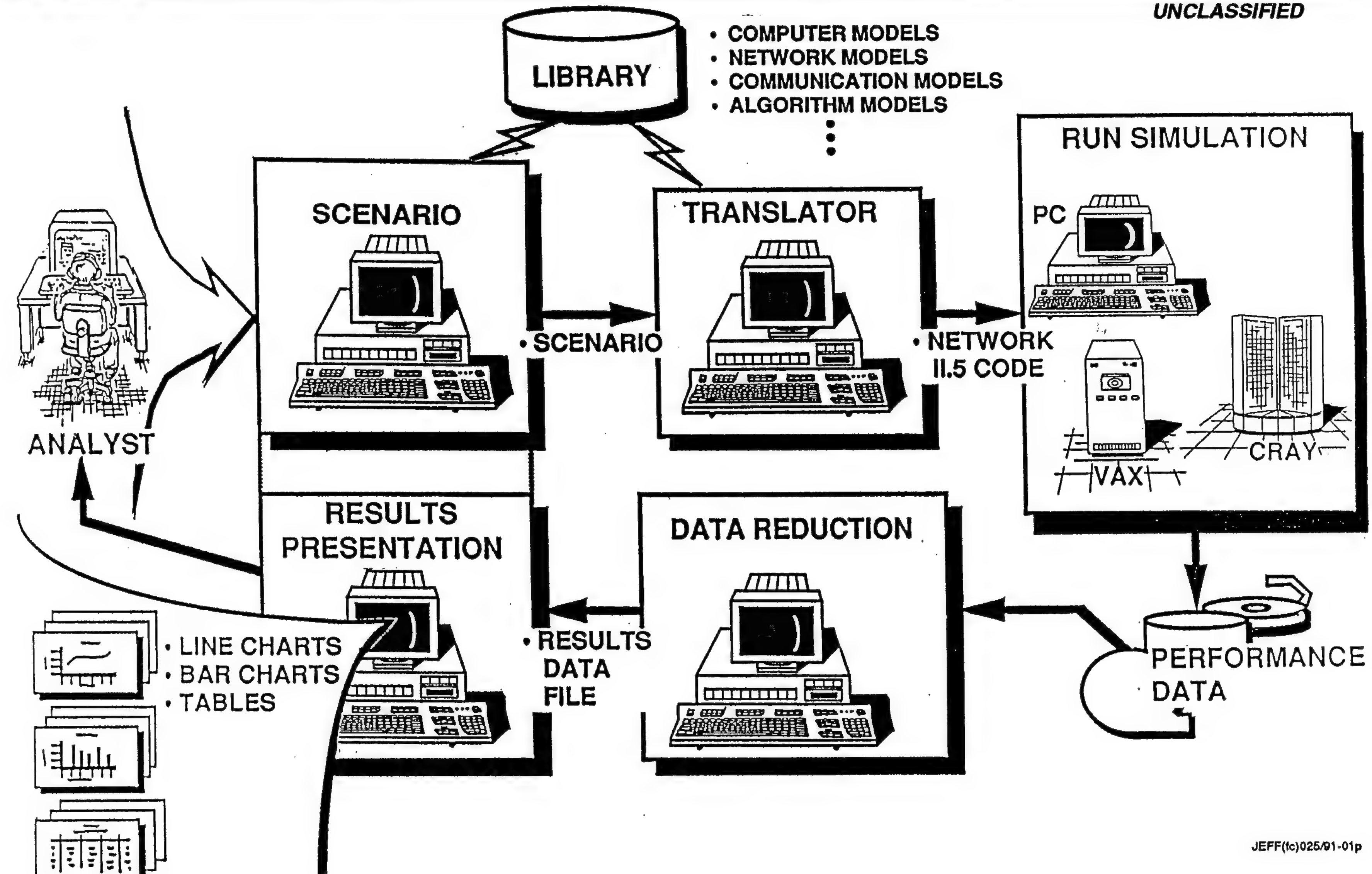
UNCLASSIFIED



TO SUPPORT RESOURCE PLANNING THROUGH GENERATION AND ANALYSIS OF MULTI-FIDELITY NETWORK SIMULATIONS

FUNCTIONAL FLOW

UNCLASSIFIED



JEFF(fc)025/91-01p

TOOL IMPROVEMENTS OVER ARCSIM (V3.0)

UNCLASSIFIED

ADDITIONAL FUNCTIONALITY

- 3 LEVELS OF DETAIL FOR COMPUTER MODEL
  - DETAILED
  - INTERMEDIATE
  - BLACK-BOX
- 3 LEVELS OF DETAIL FOR SOFTWARE TASK MODELS
  - MACHINE INSTRUCTION
  - RESOURCE USAGE
  - BLACK-BOX
- OPERATING SYSTEM MODEL
- SECONDARY MEMORY DEVICE MODELS
  - DISK UNITS
  - TAPE UNITS
- PERIPHERAL DEVICE MODELS (PRINTERS, PLOTTERS, MODEMS, etc.)
- DATABASE MODEL
- DATA CHANNEL MODELS FOR FIFO, TOKEN RING, SLOTTED TOKEN RING
- USER-SPECIFIED PATH FOR TASK-TO-TASK COMMUNICATION

TOOL IMPROVEMENTS OVER ARCSIM (V3.0)

UNCLASSIFIED

USER INTERFACE IMPROVEMENTS

- REDUCE-TO-FIT/ZOOM SCREEN DISPLAY
- KIVIAT GRAPH FOR UTILIZATION REPORTS

DATA COLLECTION METHOD IMPROVEMENTS

- AUTOMATIC GENERATION OF BENCHMARKING CODE
- AUTOMATIC REDUCTION AND ANALYSIS OF BENCHMARKING DATA
- AUTOMATIC GENERATION OF COMPUTER LIBRARY INPUT FILES

TOOL IMPROVEMENTS OVER ARCSIM (V3.0)

UNCLASSIFIED

NEW REPORTS

- USER WORKLOAD STATISTICS REPORTS
  - RESPONSE TIME
  - TURNAROUND TIME
  - REACTION TIME
  - CONNECT TIME
  - CPU TO CONNECT TIME RATIO
  - PROGRAM INTERRUPT DELAY
- MULTIPLE RUN CORRELATION
- REGRESSION PLOTS
- TWO VARIABLE TIME SERIES REPORTS
- CONFIDENCE INTERNAL ASSESSMENT REPORT
- CHI-SQUARE TEST FOR COMPARISON OF SIMULATION RESULTS TO EMPIRICAL DATA

McGRAW(fc)102/91-06a

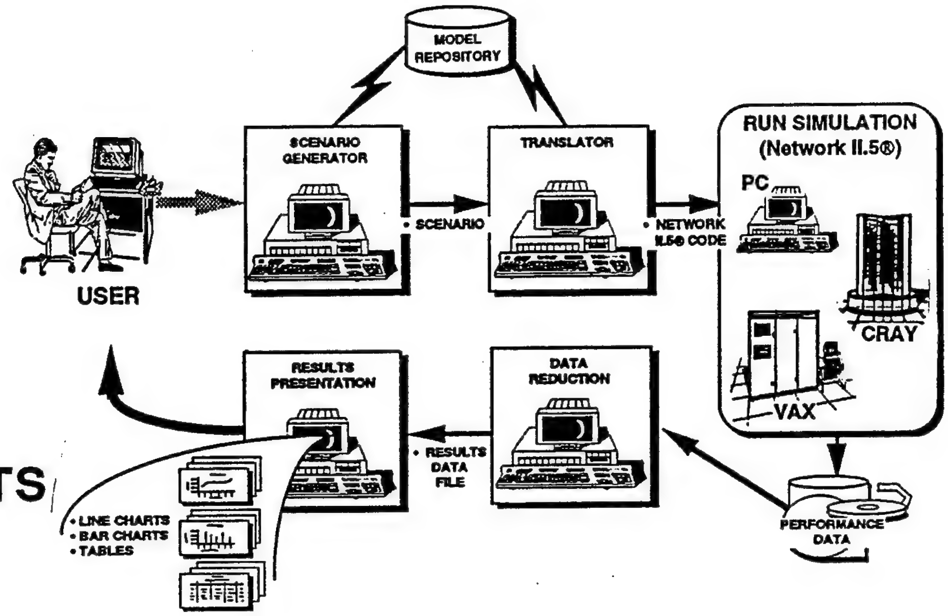
ARCSIM V3.1 BASELINE

UNCLASSIFIED

(ENHANCEMENTS)

- INTEGRATION OF NETWORK II.5 V6.0
- COMPLETION OF T1 MODEL
- EFFICIENCY ENHANCEMENTS
- PATH SELECTION FACILITY
- PRINT TEXT/PRINT GRAPHICS FACILITY

COMPUTER AIDED SIMULATION GENERATION FOR DISTRIBUTED COMPUTING ENVIRONMENTS



OVERVIEW

Optimization Technology, Inc. (OTI) has developed an advanced capability for the rapid generation of accurate high fidelity simulations of distributed computer environments composed of multi-network, multi-protocol and multi-computer configurations. This technology has been successfully applied to such diverse domains as the Strategic Defense Command's Advanced Research Center, the Strategic Defense Initiative National Test Facility and the Air Force Satellite Control Network; each of which are representative of the complex networked computing configurations that are commonplace in today's environment. Resulting simulations of these configurations support such activities as (1) optimizing connectivity and resources for specific applications, (2) identifying overloads, bottlenecks and inefficiencies in given network configurations, (3) optimizing computer/network expansion and growth and (4) optimizing resource management and configuration control. This approach to automated simulation generation of networked computing systems permits the user to develop extensive simulations in minutes or hours as opposed to weeks or months, to rapidly modify simulation functionality and to extend the domain and range of simulation capability by adding resource models to a Repository.

A fundamental and successful objective in the development of this tool was to provide individuals such as system managers, operations personnel, software designers, resource managers and network architects (not necessarily simulation experts) with the capability to rapidly generate accurate simulations of networked computer configurations. An overview of the approach is illustrated above.

OPTIMIZATION TECHNOLOGY, INC. • PROGRESS CENTER, SUITE 600 • 6767 MADISON PIKE, NW • HUNTSVILLE, ALABAMA 35806
Phone (205) 721-1288

The heart of the system is a model Repository or library containing verified computer, network, protocol, router, long haul communication, algorithm, etc. models. With assistance from the tool, models can be easily added to the repository at any level of detail or fidelity. Currently the repository contains very high fidelity models which have been validated with an empirical test suite (also provided by the tool). To enhance the tool's usefulness in the investigation of network/computer expansion options, lower fidelity models created from vendor data can be quickly developed and entered into the repository. As more accurate data are made available, the fidelity of these models can be easily increased.

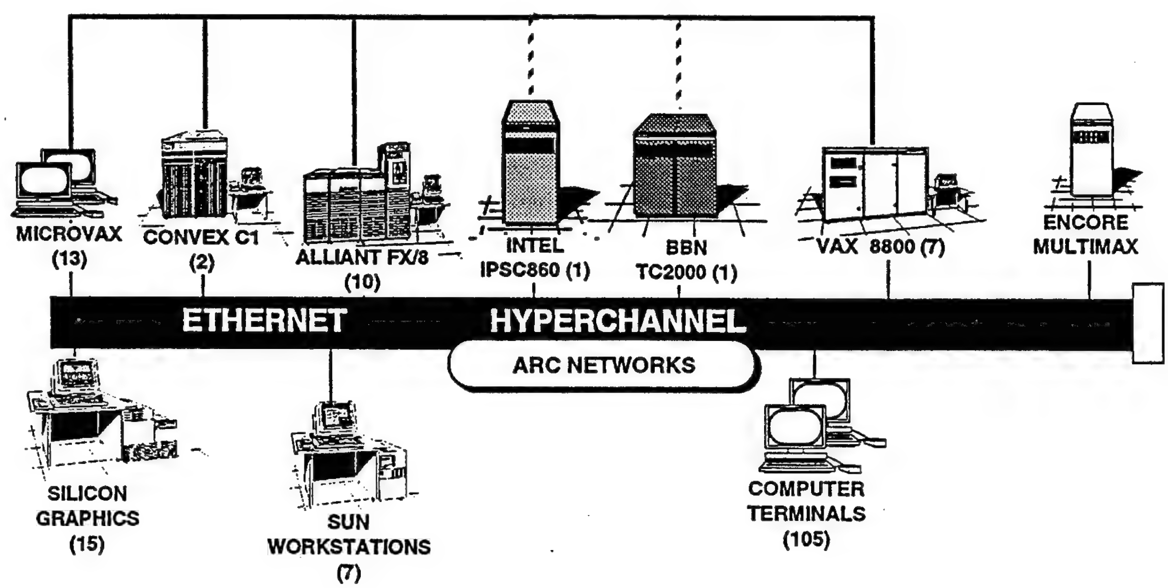
In order to construct a simulation of the desired configuration the user first graphically describes the network and network connectivity, the computers, the proposed software architecture, etc. The Scenario Generator retrieves the desig-nated models from the Repository and appropriately binds and enhances them in a form to be passed to the Translator, which in turn creates a Network II.5® image of the simulation to be generated.

Network II.5®, developed by CACI, is a powerful event driven simulation language that incorporates the ability to model complex and dynamic system interactions at varying degrees of fidelity. The presence of pre-defined building blocks such as Processing Elements, Transfer and Storage Devices, as well as some pre-defined Media Access layer protocols and detailed data collection capabilities, make Network II.5® an ideal medium for the translation stage of the tool.

The source image of the simulation is then translated to a runtime image to be executed on one of many computing platforms supported by Network II.5®. Performance data is collected during runtime and stored for subsequent data reduction. Network II.5® has an extremely powerful data collection capability, providing data on virtually any aspect of simulation performance.

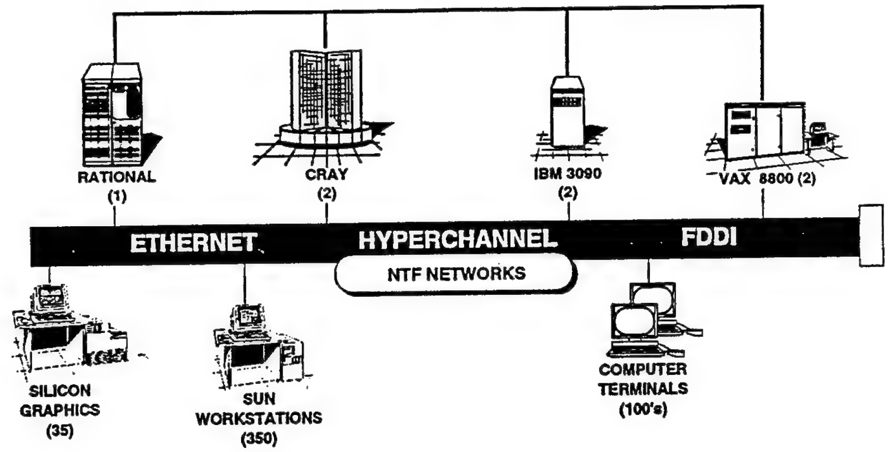
The Data Reduction facility provides a data file to the Results Presentation facility, which in turn produces line charts, bar charts, tables, etc. as well as tabular accounts of the resulting performance. The Results Presentation facility supports the presentation of user designated performance measures such as throughput, utilization, task information, device usage and media access by isolating appropriate performance entities, developing their relationships and plotting them in a pre-designated format.

OTI first developed a version of this tool, ARCSIM, under government contract for the Strategic Defense Command's Advanced Research Center (ARC). The ARC contains a massive amount of computer and networking resources as shown below. For this application, the tool is principally utilized to optimize the distributed environment for large complex defense experiments. The experiment may be configured using any subset of available resources. An experiment planner has the ability to accurately simulate and optimize the configuration and its associated performance prior to committing to a given topology.



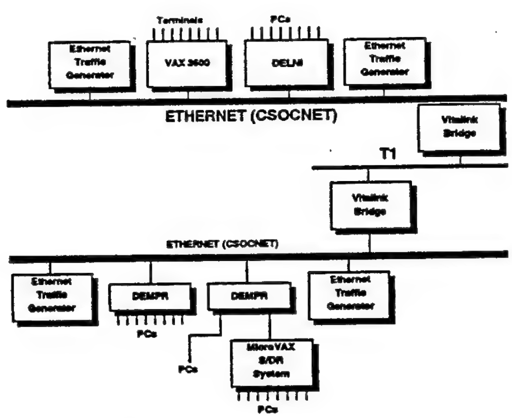
ADVANCED RESEARCH CENTER

OTI is currently developing a derivative of the ARCSIM tool for the Strategic Defense Initiative National Test Facility (NTF). This tool, called NTBSIM will not only provide the capability to support National Testbed architectural analysis through the addition of resource models to the Repository but will also provide the functionality for resource planning and management. In addition to expanding the model Repository, the Scenario Generator is being modified and the Results Presentation capability is being extended to accommodate the new capabilities. An overview of NTF resources is presented below.

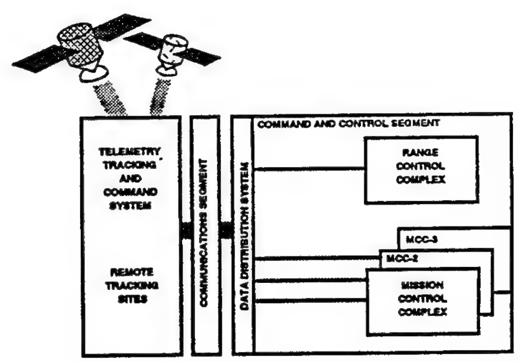


NATIONAL TEST FACILITY

OTI is developing yet another derivative of the ARCSIM tool for the Air Force Satellite Control Network to (1) support the Mission Support Network Service/Deficiency Reporting System by providing a capability for the analysis of system access delays and (2) provide an AFSCN Performance and Analysis Tool (APSAT). An overview of these two applications is presented below.



MISSION SUPPORT NETWORK
SERVICE/DEFICIENCY REPORTING SYSTEM



APSAT

The capabilities for the above three applications have been hosted on an IBM PC compatible machine, (386 with expanded memory) in Pascal. Network II.5® is available for the PC, which provides the user with a complete PC environment. As pointed out earlier, Network II.5® is also available for many other platforms. For extensive simulations, the user generates the run-time image on the PC, executes the simulation on an alternate platform, and ports the performance data back to the PC for Data Reduction and Results Presentation.

PRESENTER: Evan Lock

Re-Engineering Existing Software Into Distributed Applications

Re–Engineering Existing Software Into Distributed Applications

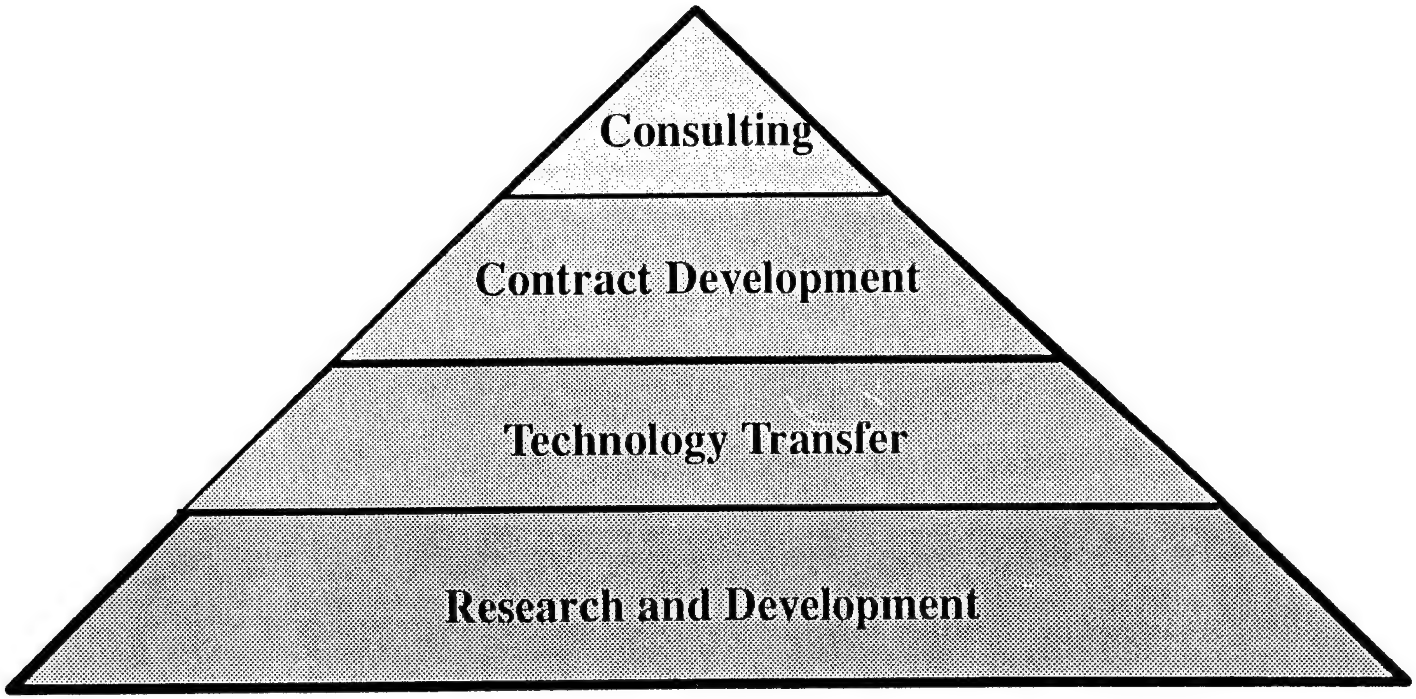
USASDC 4th Computer Resource Integration Meeting

Mr. Evan Lock
President and Chief Executive Officer
Computer Command and Control Company
2300 Chestnut Street, Ste. 230
Philadelphia, PA 19103

tel.: 215–854–0555
fax: 215–854–0665

Computer Command and Control Company

Business Focus



Develop and Apply Leading Edge Software Engineering Technology, Emphasizing Advances in the Use of Specifications and Automation.

The Re–engineering Challenge

How do I . . .

- Capture Essential Algorithms
- Generate Design Graphics
- Obtain Assistance on Design Modifications
- Address Platform Dependent Implementation
- Introduce Object Orientation
- <u>Meet</u> <u>Timing</u> <u>Deadlines</u>

Modern Applications Will Take Advantage of Distributed Architectures

If Additional Complexities Are Addressed . . .

Synchronization? *Heterogeneous Networks?*

Allocation?

Deadlock?

Timing?

Scheduling?

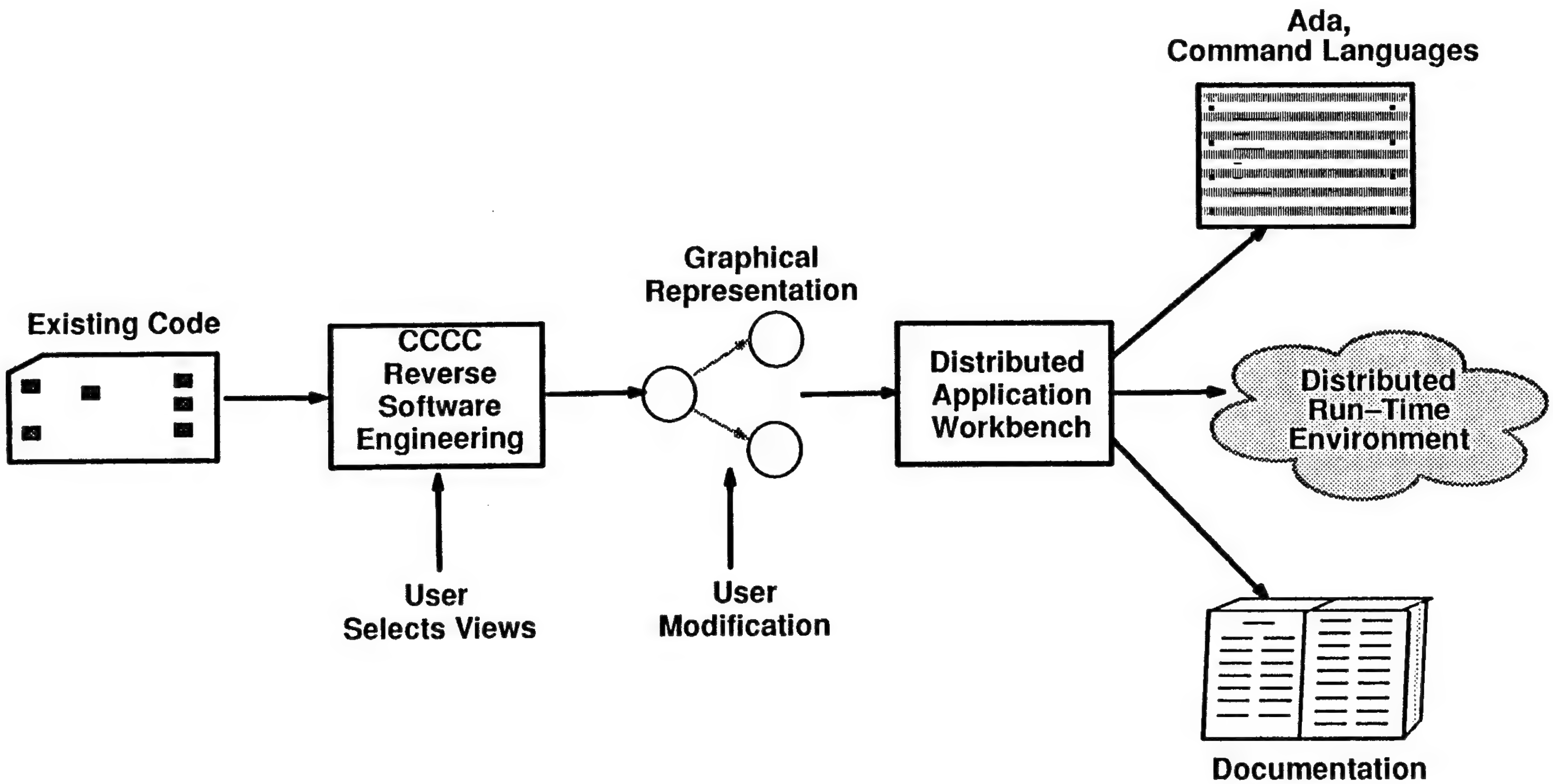
Tasking? *Mailboxes?*

Protocols?

Object Orientation?

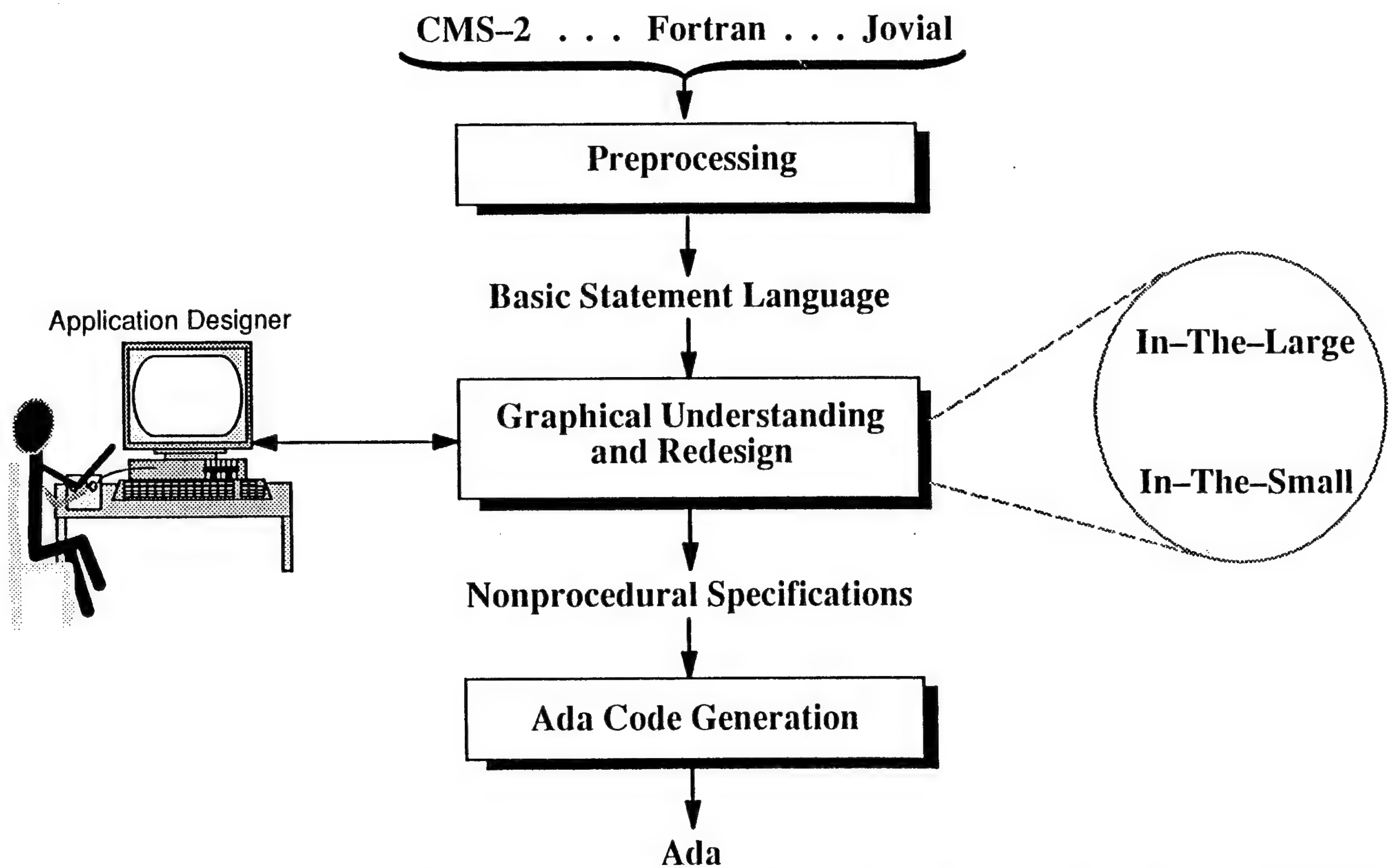
. . . And Achieve Price/Performance Advantage

Summary of Overall Approach

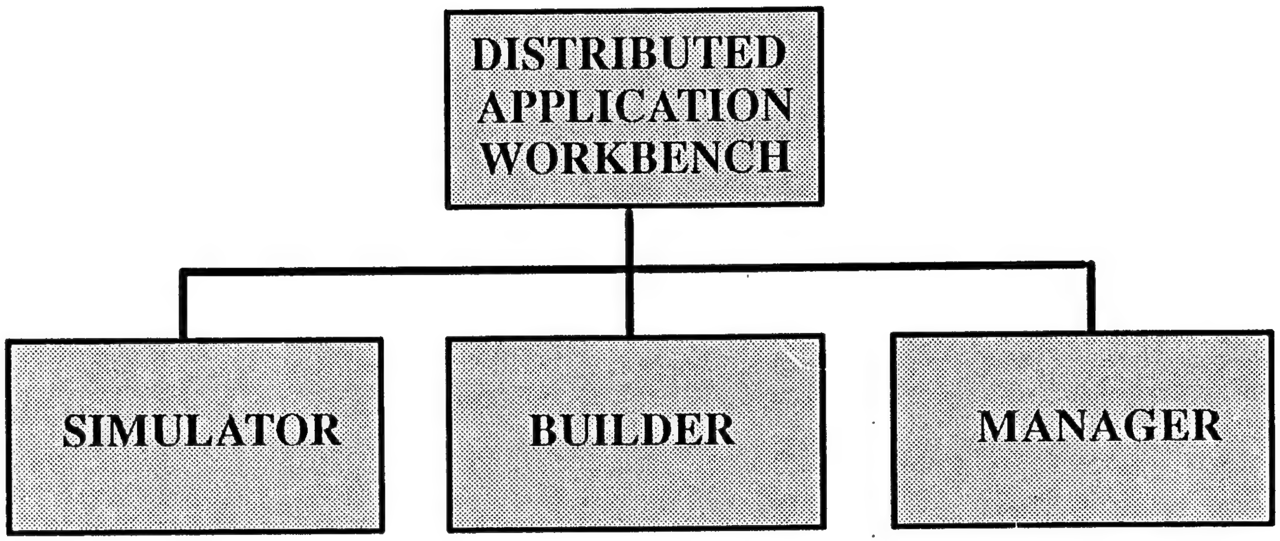


Supported by Integrated Toolset

Summary of CCCC Re–Engineering System



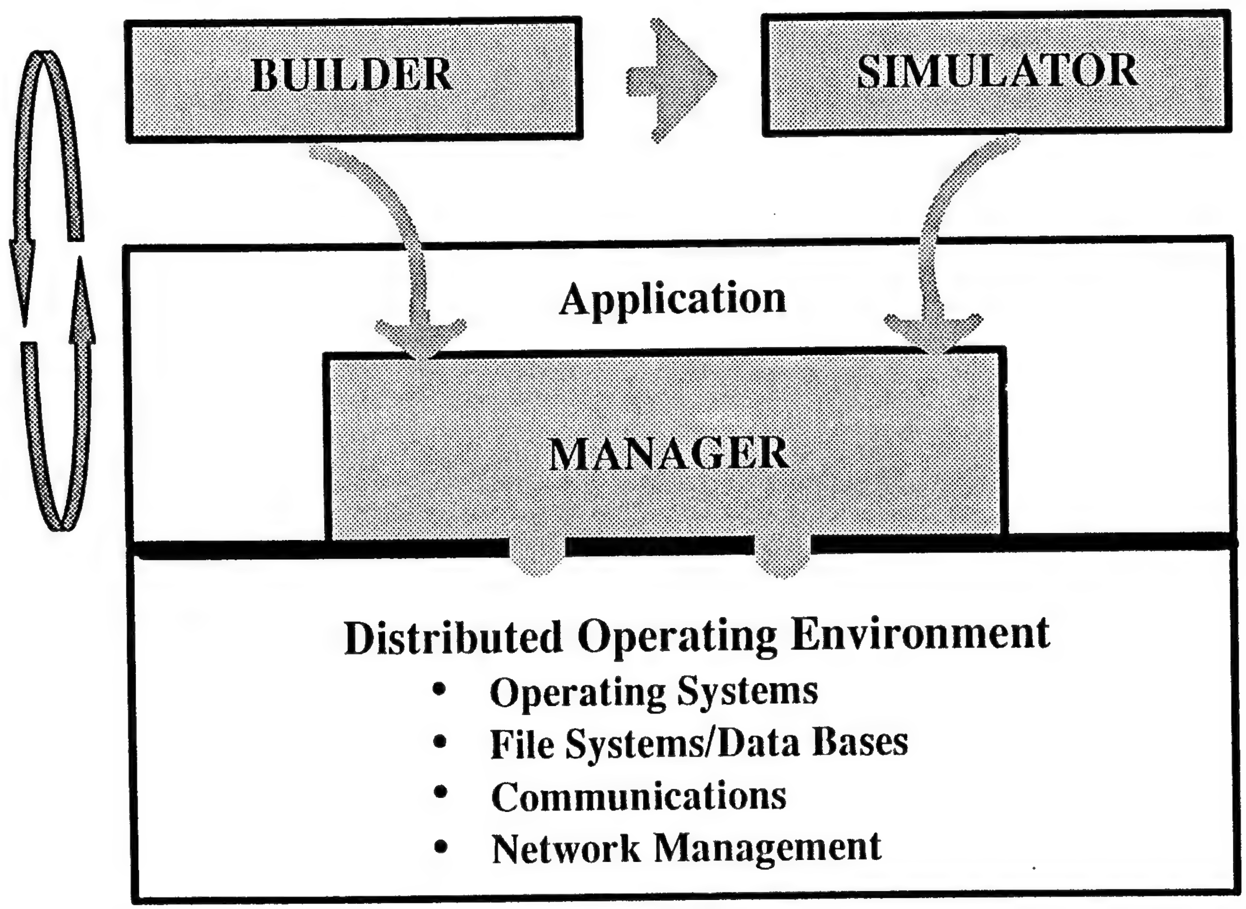
Distributed Application Workbench



A high level tool set for developing, managing and maintaining distributed applications.

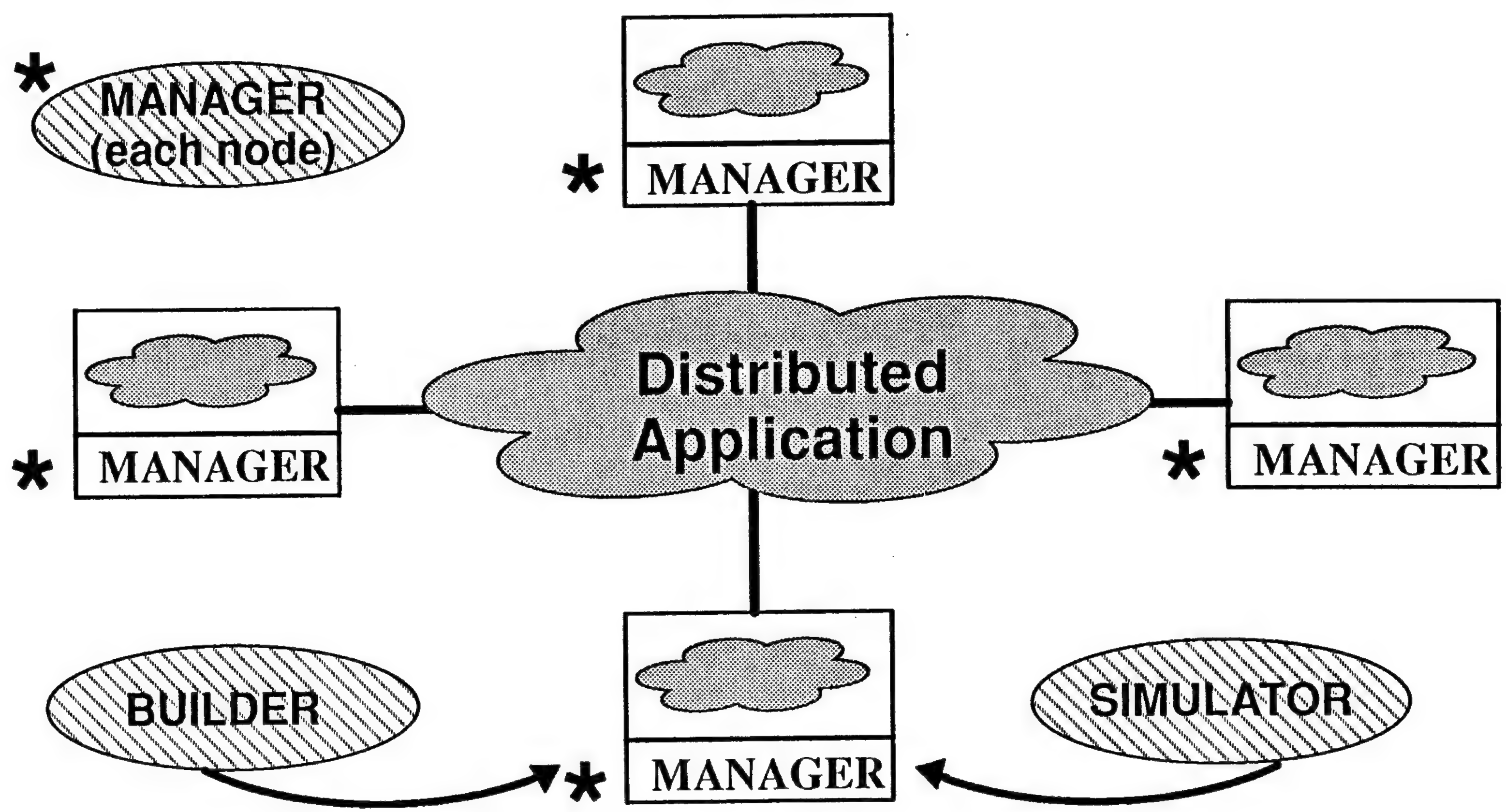
Distributed Application Workbench

Role Within Application Environment

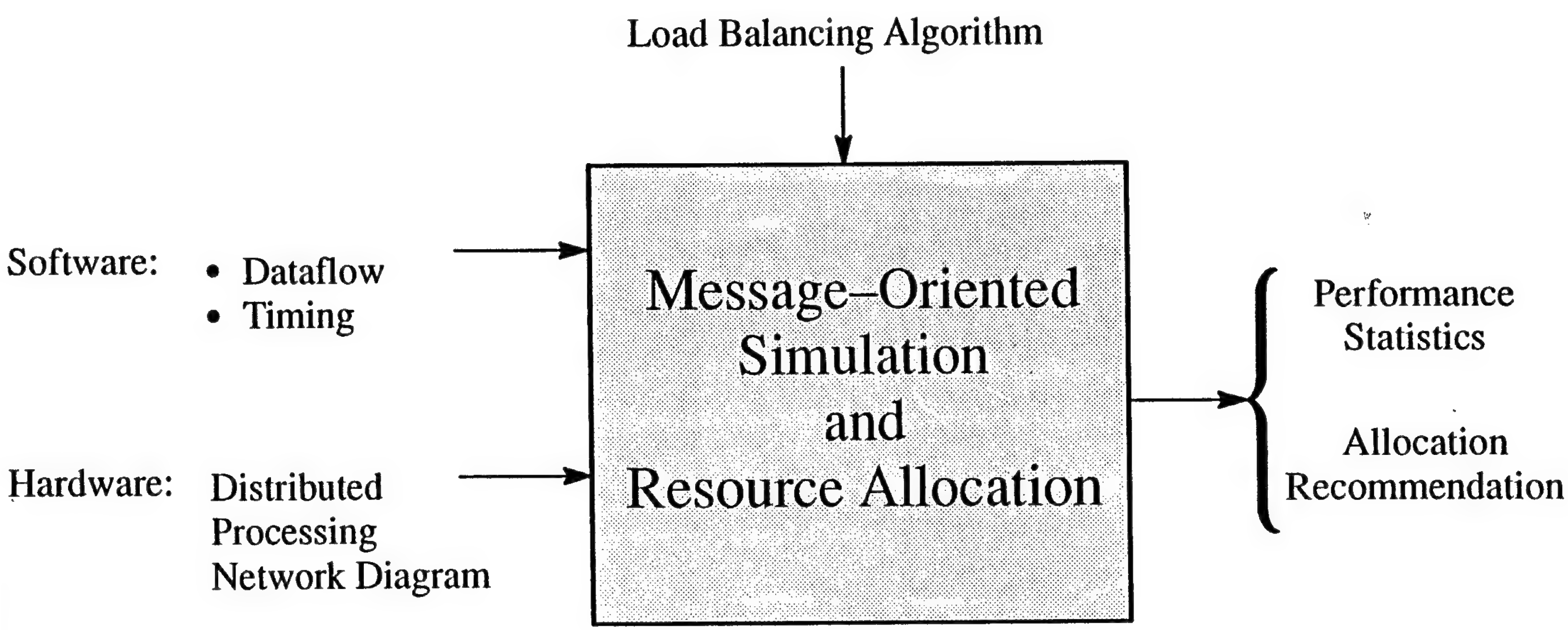


Distributed Application Workbench

Role Within Network



SIMULATOR Functionality

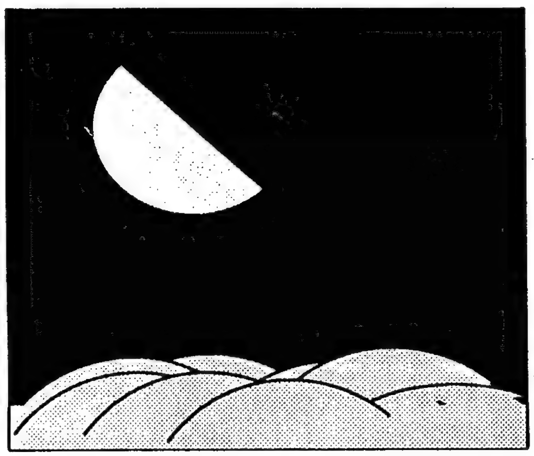


SIMULATOR Features

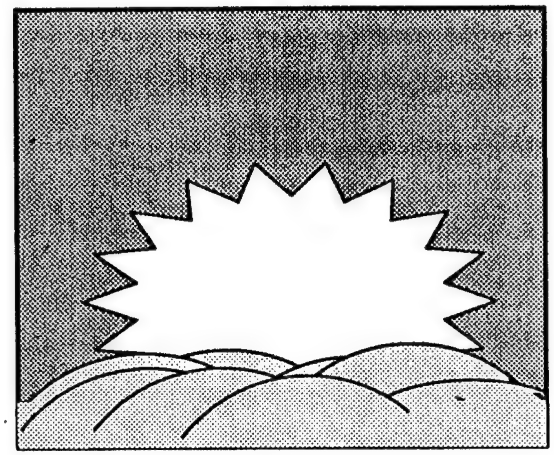
- Proposes Best Way to Map Software on to Hardware
- Provides Design Feedback
- Projects Optimum Resource Utilization
- Open to Different Scheduling and Allocation Algorithms
- Can Be Utilized at Compile Time and Run Time

SIMULATOR Benefits

Ad Hoc Approach

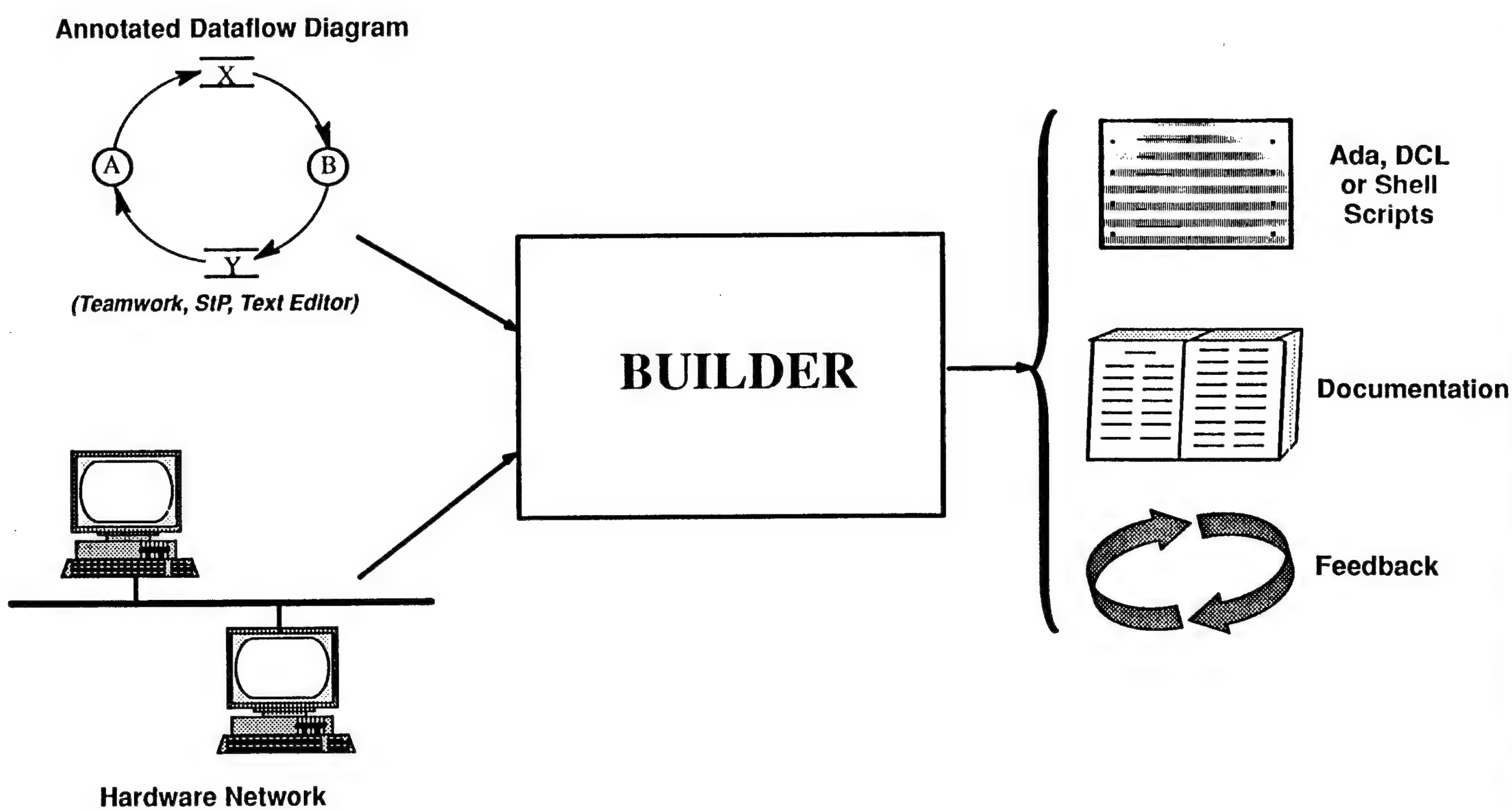


Systematic Approach



- Better Performing Application
- Return on Hardware Investment
- Application More Responsive to User's Changing Needs
- Reduces Maintenance Burden

BUILDER Functionality

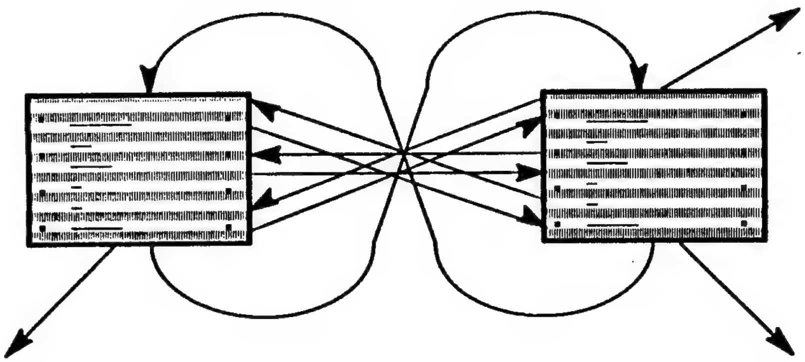


BUILDER Features

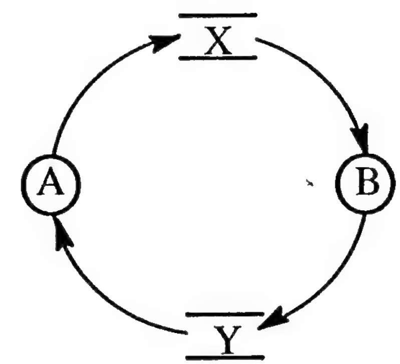
- Uses Graphical Specification Language
- Bridges Analysis and Design Activities
- Analyzes Specification for Deadlock
- Generates Programs for Multiple Environments
- Produces System Documentation
- Accepts Hierarchical Decomposition

BUILDER Benefits

Conventional Confusion

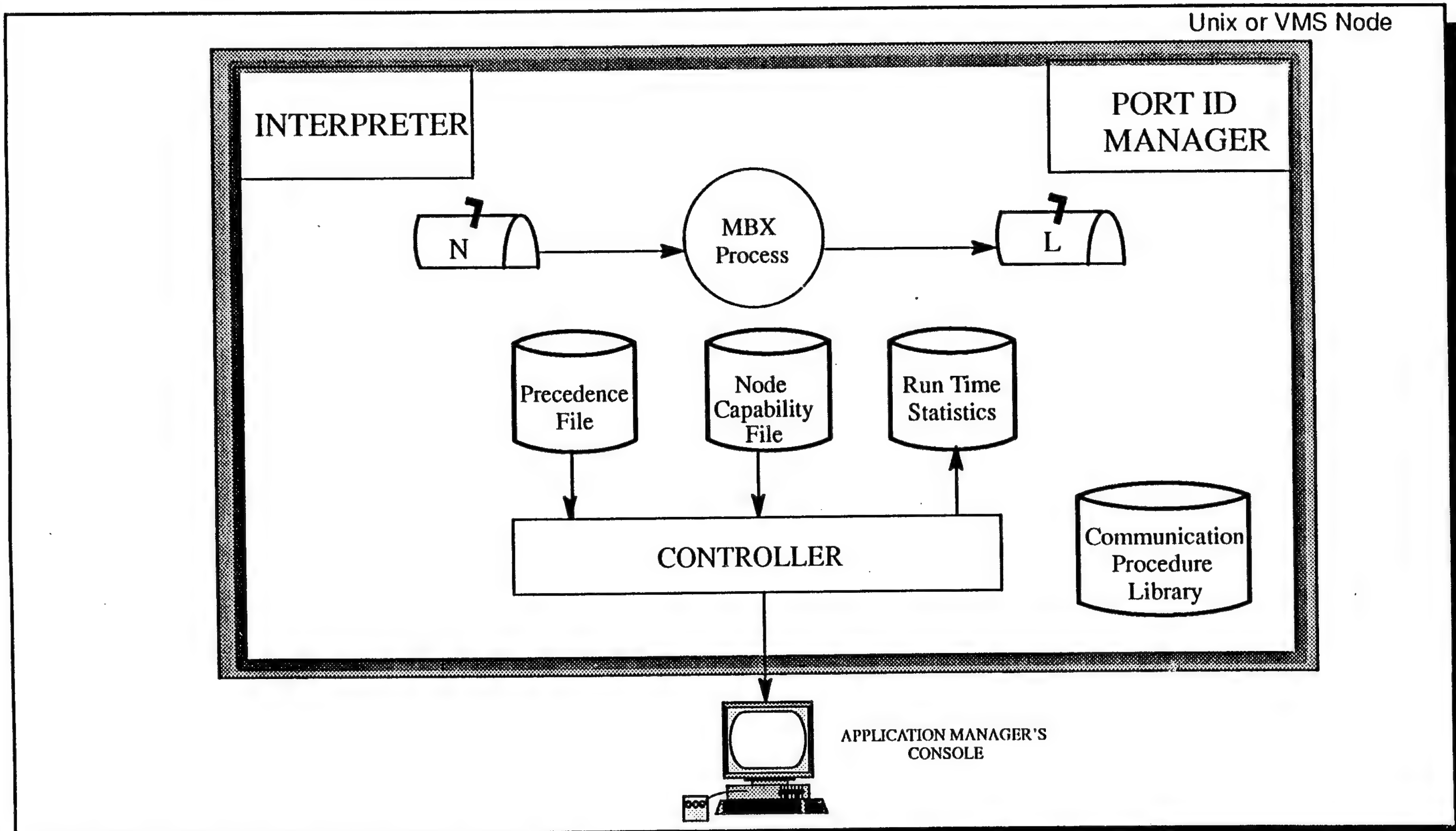


Graphical Specification



- Increase Analyst/Programmer Productivity
- Faster Response to Changing Needs
- Reduces Maintenance
- More Reliable Operation

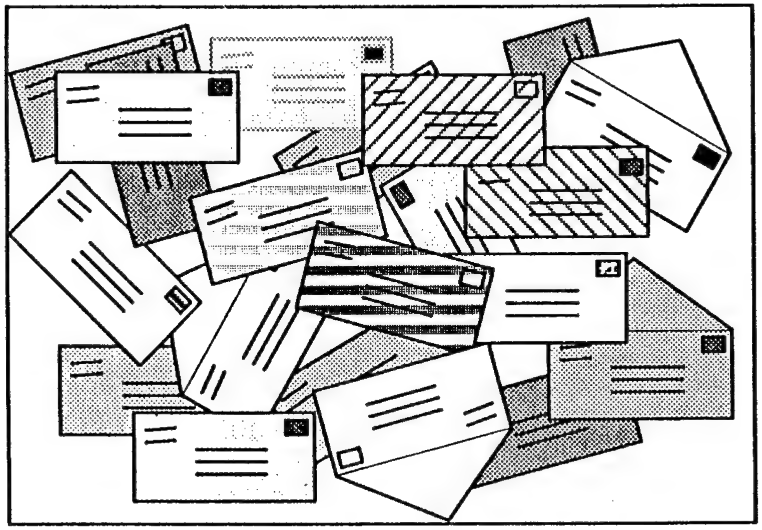
MANAGER Functionality

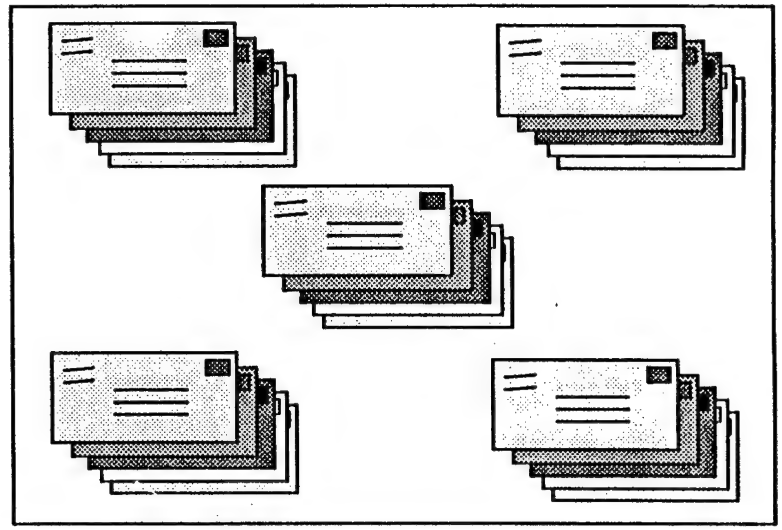


MANAGER Features

- Hides Complexities of Operating System and Communications
- Dynamically Monitors Application Performance
- Operates in Heterogeneous Environments
- Synchronizes Tasks and Communications
- Permits Dynamic Modification of Application Configuration
- Provides Run Time Logging
- Interfaces with Advisor at Run Time

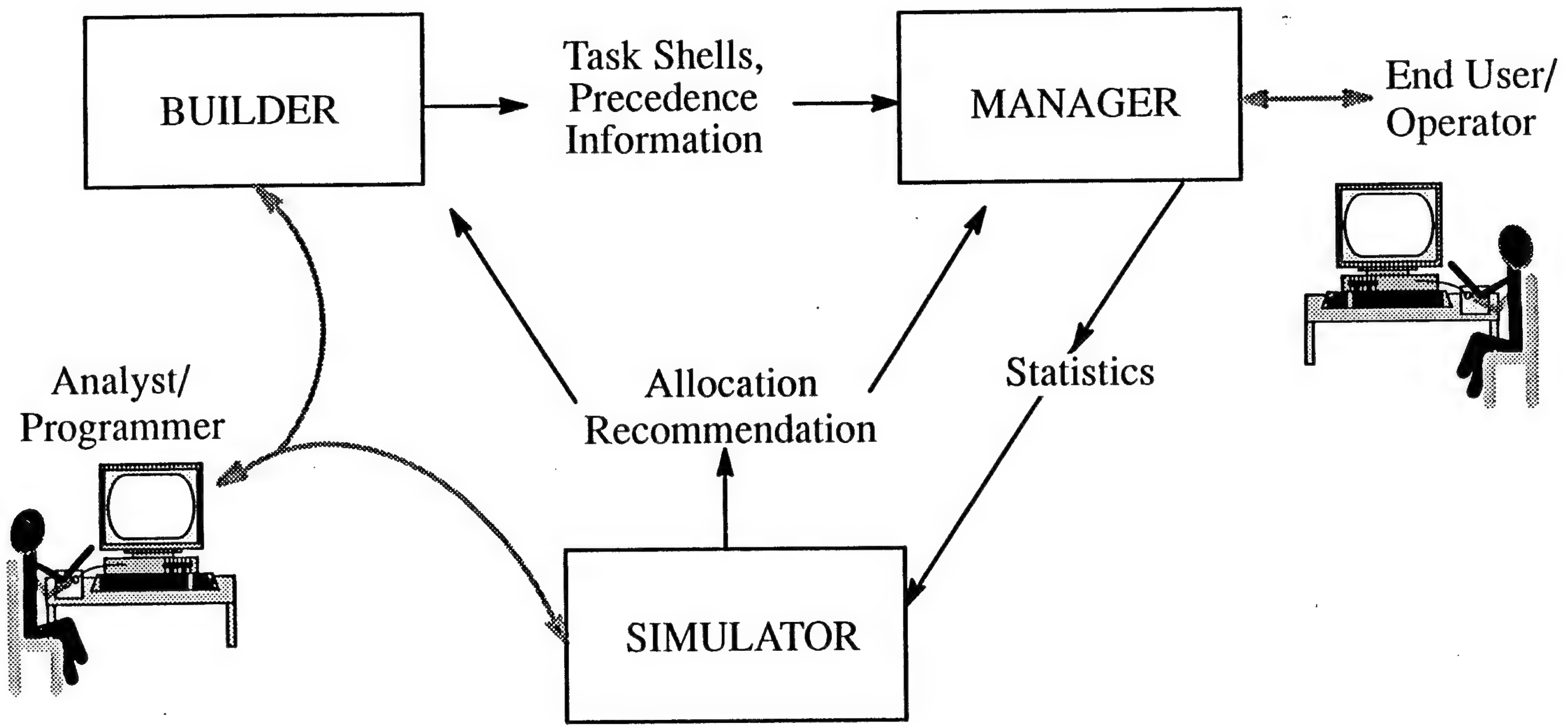
MANAGER Benefits





- Faster Response to User's Needs
- Improve Quality of Decisions
- More Efficient Use of Staff
- Better Application Performance
- Increased Return on Hardware Investment

SIMULATOR, BUILDER, MANAGER Working Together

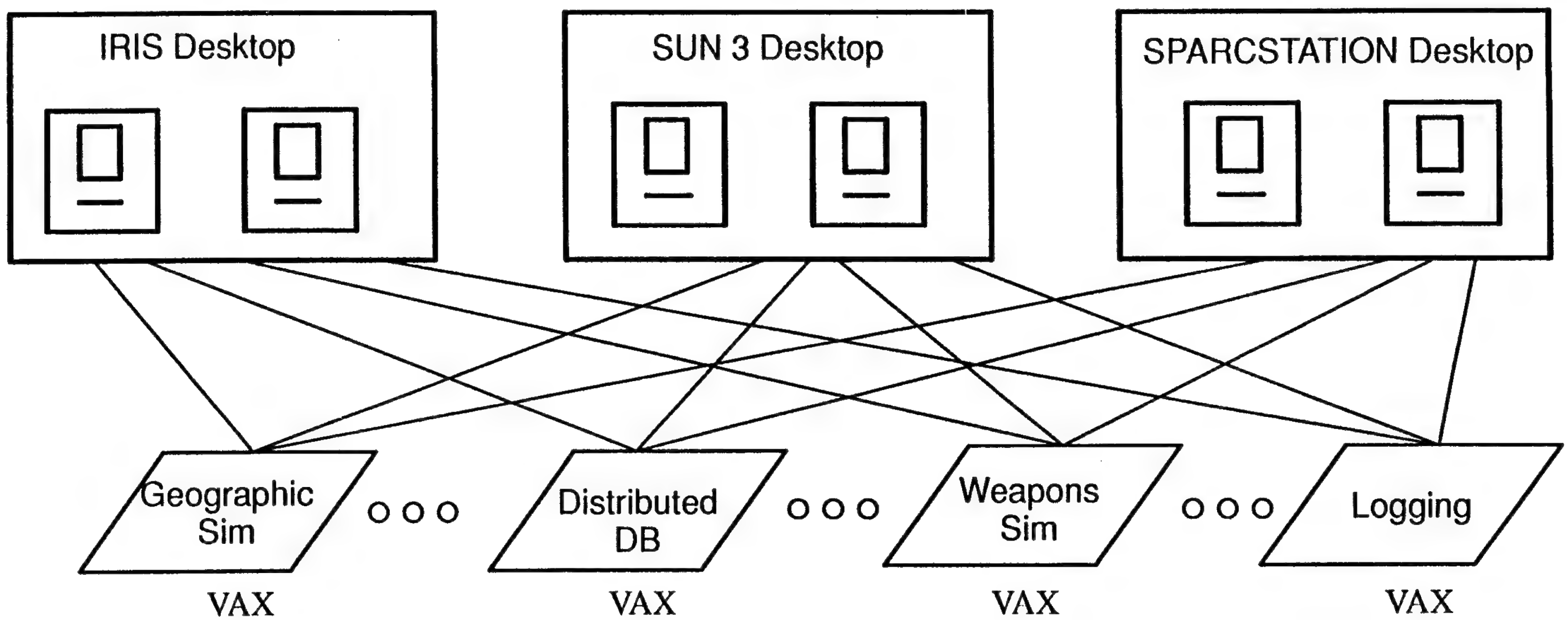


Distributed Success

Examples Using Distributed Application Workbench

Computer Command and Control Company

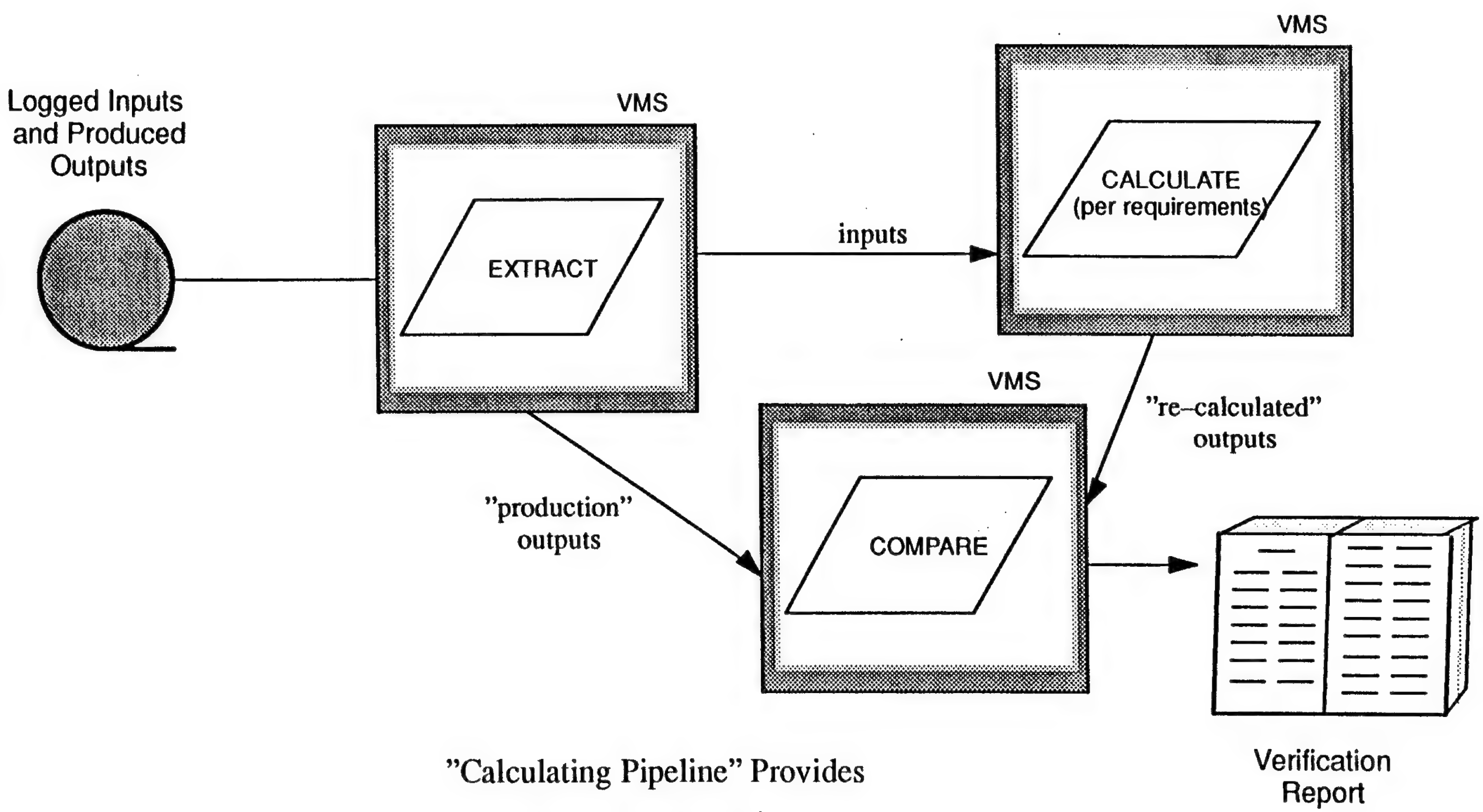
Distributed Combat Systems Simulation (NUSC)



- Increased Simulation Productivity
- Increased Operational Reliability
- Higher Performance Design for Adverse Battle Condition Response

Computer Command and Control Company

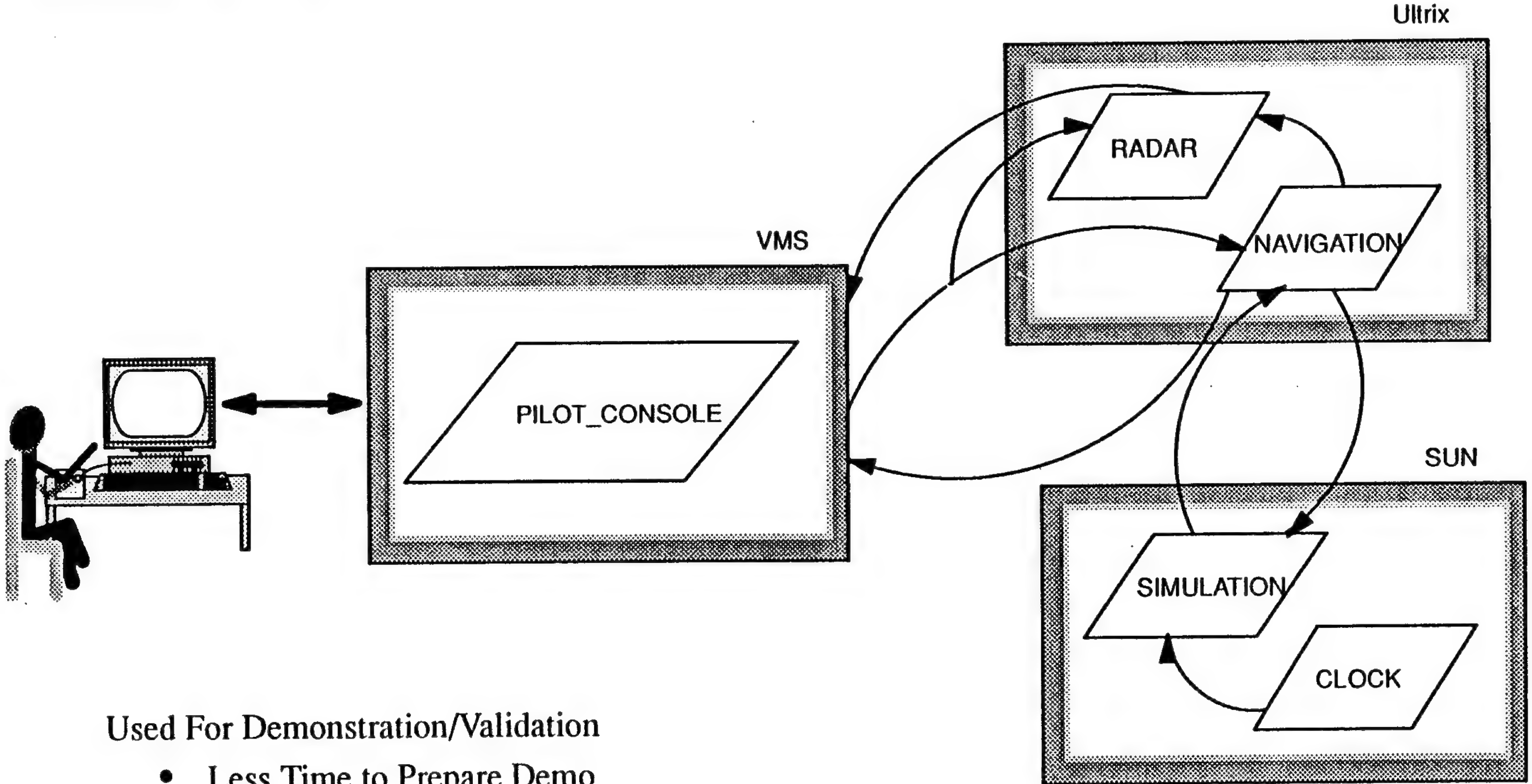
Distributed Verification Calculations (DOD Contractor)



"Calculating Pipeline" Provides

- Faster Turnaround
- Increased Resource Utilization

Distributed Avionics Simulation – LAMPS Helicopter (NADC)



Used For Demonstration/Validation

- Less Time to Prepare Demo
- More Accurate Demo/Validation – Improved Design

Distributed Application Workbench

Benefits Summary

- Stimulate New Business Opportunities
  - More Flexible System Rapidly Adapts to Change
  - Reduce Cost/Disruption from Change
- Increase Return on Investment
  - Current System
  - Idle Workstations
- Improve Quality of Decisions
- Reduce Application Maintenance Costs
- Less Investment in Critical Personnel
- Boost Productivity Throughout Software Life Cycle

PRESENTER: Gregory H. Chisholm

SDIO Related Activities at Argonne

SDIO Related Activities at Argonne

Gregory H. Chisholm

Argonne National Laboratory

May 13, 1991

Outline for Presentation

- ☐ Software Engineering Environment and Tools
- ☐ Parallel Simulation
- ☐ Fault-tolerant, Reliable, Portable Computing for the SDS

Software Engineering Environment and Tools

Goal: to assist SDI contractors in learning to use parallel processors and to provide them with access to parallel machines of a wide range of architectures

- ☐ Access to Parallel Machines: ACRF Operation
- ☐ Parallel Programming Classes in Ada, FORTRAN and C
- ☐ Development of portable parallel-programming tools; ANL Macros, p4 and PARMACS.

Software Engineering Environment and Tools (Cont'd)

Access to Parallel Machines — Advanced Computing Research Facility (ACRF)

- ☐ Currently the ACRF includes nine commercial multiprocessors:
  - An Alliant FX/8 with 8 vector processors sharing 64 Mbytes of memory.
  - A Sequent Symmetry 81 with 26 processors sharing 32 Mb of memory.
  - An Encore Multimax with 20 processors sharing 64 Mb of memory.
  - An Active Memory Technology DAP-510 with 1024 one-bit processors (8KB each).

Software Engineering Environment and Tools (Cont'd)

Access to Parallel Machines — ACRF

- ACRF multiprocessors (Cont'd):
  - A Thinking Machines Connection Machine 2 with 16384 one-bit processors (128 Mb total).
  - An Ardent Titan-4 with 4 vector processors sharing 32 Mb of memory and high-performance graphics.
  - A BBN Butterfly TC2000 with 45 processors sharing 192 Mb of memory.
  - An Intel iPSC/860 8 node hypercube with 16 Mb of memory per node.
  - The "world's largest and fastest" computer, an Intel iPSC/860, 528 node hypercube with 8.5 GB total memory. Acquired jointly by CalTech, ANL and others.

Software Engineering Environment and Tools (Cont'd)

The ANL ACRF

☐ Purpose

- Carry out research in parallel processing
- Provide facilities for research by national/international user community
- Educate users
- Transfer technology among universities, industry, and laboratories

Software Engineering Environment and Tools (Cont'd)

ACRF Activities

- ☐ Encourage research by members of MCS and other divisions at Argonne
- ☐ Operate facility with a variety of advanced architecture computers
- ☐ Provide a testbed for new machines in mult-user, multi-application setting
- ☐ Provide support for outside research through software and documentation
- ☐ Educate potential users through classes, seminars, and workshops

Software Engineering Environment and Tools (Cont'd)

Parallel Programming Classes

- ☐ Offered for following languages:
  - Ada
  - FORTRAN
  - C
- ☐ Ada compilers on parallel machines
  - Sequent — Symmetry
  - Encore — Multimax
- ☐ Ada compilers on sequential machines
  - Sun workstations

Software Engineering Environment and Tools (Cont'd)

Tools

- ☐ Development of portable parallel-programming tools:
  - shared-memory tools
  - message-passing tools
  - program transformations
- ☐ Parallel Logic Programming
  - Strand
  - Aurora Parallel Prolog

Parallel Simulation

Goal: to develop approaches for parallelizing SDI algorithms and to identify the optimal environment for running these algorithms

- ☐ Are heterogeneous parallel architectures feasible for SDI battle management computation?
- ☐ If one looks at typical SDI BM algorithms, different algorithms achieve their best performance on different architectures
  - Accessibility achieves highest performance on Single Instruction Multiple Data (SIMD)
  - Weapon Target Assignment (WTA) performs best on shared memory machines

Parallel Simulation (Cont'd)

- Background
  - ANL has parallelized a number of SDI algorithms on a variety of architectures, and
  - has developed techniques for distributing computation over networks
- Plan — Conduct an experiment that determines performance of computing SDI functions on a network of diverse architectures

Parallel Simulation (Cont'd)

☐ Potential Results:

- Feasibility/infeasibility of heterogeneous architectures using COTS networking and I/O
- Identify I/O bottlenecks for future investigation if heterogeneous architecture is required — how much improvement in COTS networking would be required?
- Identify need for new algorithms for particular parallel processor types if heterogeneous architecture not feasible or not cost-effective

Fault-tolerant, Reliable, Portable Computing for the SDS

Goal: to explore new technologies in support of the design of SDS functions and to enhance the environment for writing portable and parallel Ada software

- ☐ Fault-tolerant Parallel Processing Experiment
- ☐ Accurate, Portable, Standardized Ada Math Libraries

Fault-tolerant, Reliable, Portable Computing for the SDS (Cont'd)

Fault-tolerant Parallel Processing Experiment

- Goal: demonstrate that representative SDS computations can be executed in parallel on fault tolerant hardware
- Fault-tolerant, safe, secure and parallel, can one system do it all?
- What is the cost in efficiency when these requirements are coalesced into a single implementation?

Fault-tolerant, Reliable, Portable Computing for the SDS (Cont'd)

Fault-tolerant Parallel Processing Experiment (Cont'd)

- ☐ Plan — Conduct an experiment to investigate these questions
- ☐ Elements of the experiment:
  - Parallel, fault-tolerant computing (CSDL FTPP, ANL)
  - SDS algorithm investigations (Alphatech, Inc., ANL)
  - Early IV&V (ANL, MSU, UNM)
  - Security (UofC, Santa Barbara)
  - Safety (ANL, Geo. Mason U)

Fault-tolerant, Reliable, Portable Computing for the SDS (Cont'd)

Accurate, Portable, Standardized Ada Math Libraries

- ☐ Observation: it is difficult to write portable numerically computations in Ada because there are no standard numeric packages
- ☐ Goal: enhance environment for writing portable, numerically intensive Ada software
- ☐ Plan — develop theory, practice, testing techniques, and standards for numeric packages in Ada

Fault-tolerant, Reliable, Portable Computing for the SDS (Cont'd)

Accurate, Portable, Standardized Ada Math Libraries (Cont'd)

- ☐ Package of elementary functions (sin, cos, exp, tan, etc.)
  - Conforms to proposed standard
  - Includes all 29 functions
  - Satisfies accuracy requirements
- ☐ Implementation is portable
  - Compiles without source code modification on various environments
  - Consistently accurate on different environments

Fault-tolerant, Reliable, Portable Computing for the SDS (Cont'd)

Accurate, Portable, Standardized Ada Math Libraries (Cont'd)

- ☐ Current focus — complex functions (e.g. FFT's and circuit simulation)
- ☐ Future work includes:
  - Conducting error analysis
  - Devising accurate algorithms
  - Preparing sample implementations

Fault-tolerant, Reliable, Portable Computing for the SDS (Cont'd)

Fault-tolerant Parallel Processing Experiment—Summary

- ☐ Actually, the designer of life-critical, computer-based systems has only two worries:

  1. The system *is not* operating as specified, and
  2. the system *is* operating as specified.

- ☐ *All* approaches to V&V are dependent on demonstrating compliance with a specification

→ the process is only as good as the specification.

Fault-tolerant, Reliable, Portable Computing for the SDS (Cont'd)

Fault-tolerant Parallel Processing Experiment (Summary — Cont'd)

- Properties desired in a formal specification:
  - Is complete (WRT essential properties)
  - Is unambiguous
  - Documents functional decomposition (hierarchical)
  - Provides a bridge between people and systems
  - Encompasses the totality of the system

Fault-tolerant, Reliable, Portable Computing for the SDS (Cont'd)

Fault-tolerant Parallel Processing Experiment (Introduce Case Study)

- ☐ One approach to fielding a life-critical system:
  - Determine essential properties, e.g., fault tolerance or security.
  - Verify that these properties permeate the design structure.
- ☐ How?

  OK Develop hierarchical specification that formally describes function decomposition from concept to implementation.

  $\neg$OK Design the system and formally demonstrate that the design is correct.

Fault-tolerant, Reliable, Portable Computing for the SDS (Cont'd)

Fault-tolerant Parallel Processing Experiment (Intro — Cont'd)

☐ Categories of specification requirements:

- functional requirements
  - → define functionality of system and components
  - → decomposable from high-level abstraction to components
- design requirements
  - → define constraints imposed by environment or application
  - → thread through hierarchy from top to bottom

Fault-tolerant, Reliable, Portable Computing for the SDS (Cont'd)

Fault-tolerant Parallel Processing Experiment (Example)

- ☐ Specified property — fault tolerance
  - functional requirement

    → multiple channels
  - design requirement

    → independence between channels

Fault-tolerant, Reliable, Portable Computing for the SDS (Cont'd)

Fault-tolerant Parallel Processing Experiment (Cont'd)

- Features of a hierarchical specification structure:
  - breadth — *total system*
  - depth — high-level abstraction through implementation level details

Fault-tolerant, Reliable, Portable Computing for the SDS (Cont'd)

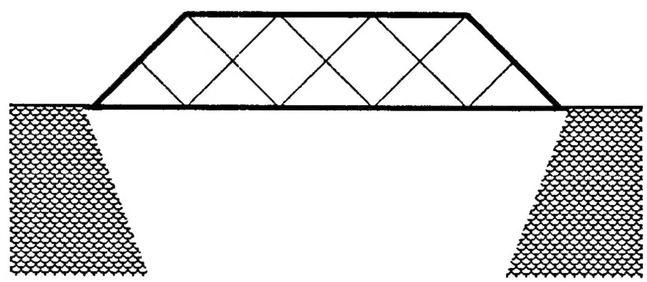
Fault-tolerant Parallel Processing Experiment (Cont'd)

- An example of a design for a life-critical controller:
  - A fault-tolerant computational platform
  - A proof that this platform is incapable of injecting faults into the running application
  - Construction of software that is proven correct WRT essential system properties*

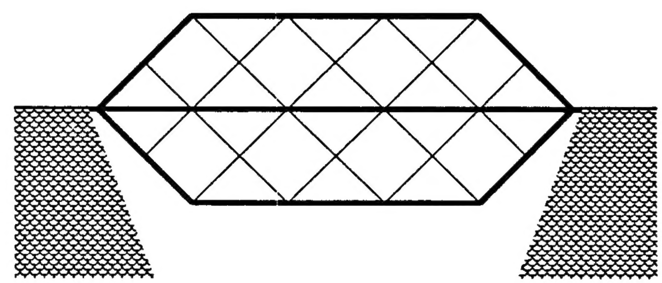
* That is, has been formally developed and analyzed to show that it correctly implements the behavior described by the system specification.

Fault-tolerant, Reliable, Portable Computing for the SDS (Cont'd)

Fault-tolerant Parallel Processing Experiment (Conclusions — Cont'd)



Tailored Formalism - Simplistic Representation Language

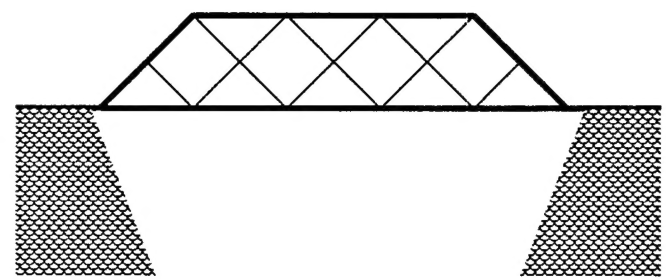


Tailored Formalism - Universal Representation Language

- ☐ Elements of bridge:
  - Structure — formalism
  - Traffic — communication
  - Form — representation

Fault-tolerant, Reliable, Portable Computing for the SDS (Cont'd)

Fault-tolerant Parallel Processing Experiment (Conclusions — Cont'd)



- ☐ A specification provides a "bridge" between people and systems.
- ☐ To provide an effective linkage, it must "eschew obfuscation."

Conclusion

- ☐ Parallel simulation — heterogeneous architecture experiment
- ☐ Fault-tolerant, Reliable, Portable Computing for SDS
  - Fault-tolerant Parallel Processing Experiment
  - Accurate, Portable, Standardized Ada Math Libraries
- ☐ Software Engineering Environment and Tools
  - ACRF — Access to parallel machines for SDI community
  - Classes in parallel programming